

# mac-address-table static

To add a list of interfaces and associated MAC addresses to the Layer 2 forwarding table, use the **mac-address-table static** command. To delete the list, use the **no** form of this command.

```
[no] mac-address-table static interface_name mac
```

## Syntax Description

<i>interface_name</i>	Specifies the interface name.
<i>mac</i>	Source MAC address in <i>aabbcc.ddeeff.gghhii</i> form.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: transparent mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **mac-address-table static** command allows you to enter static MAC addresses into the Layer 2 forwarding table. You can enter the **mac-address-table static** command multiple times with the same *interface\_name* argument to group a set of MAC addresses.

The **clear mac-address-table interface\_name** command allows you to remove the **mac-address-table** interface entries from the Layer 2 forwarding table.

The **show mac-address-table static** command allows you to display only the static MAC entries on the Layer 2 forwarding table.

## Examples

This example shows how to configure a list of interfaces and MAC addresses:

```
fwsM/context_name(config)# mac-address-table static inside 5678.aeb0.4325
Added <5678.aeb0.4325> to the bridge table
```

```
fwsM(config)# show mac-address static
interface          mac address          type      Age(min)
-----
inside             0000.0bff.0000      static
```

## Related Commands

**clear mac-address-table**  
**mac-address-table aging-time**  
**show mac-address-table**

## mac-address-table aging-time

To specify the aging time for the bridge timeout value in the Layer 2 forwarding table, use the **mac-address-table aging-time** command. To remove the bridge timeout value from the configuration, use the **no** form of this command.

[no] **mac-address-table aging-time** *minutes*

**no mac-address-table aging-time**

### Syntax Description

<i>minutes</i>	Specifies the bridge timeout aging time period in minutes.
----------------	--

### Defaults

The timeout is 5 minutes.

### Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: transparent mode

### Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

The **clear mac-address-table aging-time** command allows you to remove the bridge timeout aging time value.

### Examples

This example shows how to configure the bridge timeout aging time:

```
fwsml/context_name(config)# mac-address-table aging-time 5
```

### Related Commands

```
clear mac-address-table
mac-address-table static
show mac-address-table
```

# mac-learn

To acquire a list of MAC addresses per interface, use the **mac-learn** command. To delete the list, use the **no** form of this command.

**[no] mac-learn interface\_name disable**

## Syntax Description

*interface\_name* Specifies the interface name.

**disable** Disables MAC learning on the specified interface.

## Defaults

Enabled

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system and context command line

Command Mode: configuration mode

Firewall Mode: transparent mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **clear mac-learn** command allows you to disable the MAC address learning from all of the interfaces.

The **show mac-learn** command allows you to display the status of the MAC address learning feature on all of the interfaces.

## Examples

This example shows how to learn MAC addresses, display the results, and then clear the MAC learning:

```
fwsM/context_name (config) # mac-learn soccer

fwsM#/(config) # show mac-learn
interface                               mac learn
-----
inside                                   enabled
outside                                   enabled

fwsM(config) # clear mac-learn
Enabling learning on inside
Enabling learning on outside
```

## Related Commands

**clear mac-learn**  
**show mac-learn**

## match (route map submenu)

To define the conditions for redistributing routes from one routing protocol into another, use the **match** command in the route-map submenu. To restore the default settings, use the **no** form of this command.

```
match [interface interface_name | metric metric_value | ip address acl_id | route-type {local |
      internal | [external [type-1 | type-2]]} | nssa-external [type-1 | type-2] | ip next-hop acl_id |
      ip route-source acl_id]
```

### Syntax Description

<b>interface</b> <i>interface_name</i>	(Optional) Name of the interface.
<b>metric</b> <i>metric_value</i>	(Optional) Metric value; valid values are from 0 to 2147483647.
<b>ip-address</b> <i>acl_id</i>	(Optional) Name of an ACL.
<b>route-type local</b>	(Optional) Specifies routes that are local to a specified autonomous system.
<b>route-type internal</b>	(Optional) Specifies routes that are internal to a specified autonomous system.
<b>route-type external</b>	(Optional) Specifies routes that are external to a specified autonomous system.
<b>type-1   type-2</b>	(Optional) Specifies the type of Open Shortest Path First (OSPF) metric routes that are external to a specified autonomous system.
<b>nssa-external</b>	(Optional) OSPF metric type for routes that are external to a not-so-stubby area (NSSA).
<b>ip next-hop</b>	(Optional) Indicates where to output packets that pass a match clause of the route map.
<b>ip route-source</b>	Specifies the redistributed routes that have been advertised by routers and access servers at the address that is specified by the <i>acl_id</i> .

### Defaults

The default is **type-2**.

### Command Modes

Security Context Mode: single context mode

Access Location: system and context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

The **match ip next-hop** and **match ip route-source** commands can accept more than one *acl\_id*; they accept *acl\_id* [...*acl\_id*].

---

**Examples**

This example shows how to define the redistributed routes:

```
fws(config-route-map)# match interface soccer ip next-hop 77
```

---

**Related Commands**

**ip local pool**  
**match (route map submenu)**  
**match interface (route map submenu)**  
**match ip next-hop (route map submenu)**  
**match ip route-source (route map submenu)**  
**match metric (route map submenu)**  
**route-map**  
**set metric (route map submenu)**  
**set metric-type (route map submenu)**

## match interface (route map submode)

To distribute any routes that have their next hop out one of the interfaces specified, use the **match interface** command in the route-map submode. To remove the match interface entry, use the **no** form of this command.

[no] **match interface** *interface-name*

### Syntax Description

<i>interface-name</i>	Name of the interface.
-----------------------	------------------------

### Defaults

No match interfaces are defined.

### Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

An ellipsis (...) in the command syntax indicates that your command input can include multiple values for the interface-type interface-number arguments.

The **route-map global configuration** command and the **match** and **set route-map** configuration commands allow you to 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—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular redistribution actions to perform if the criteria that is 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 give the **match** commands in any order. All **match** commands must “pass” to cause the route to be redistributed according to the set actions that are given with the **set** commands. The **no** forms of the **match** commands remove the specified match criteria.

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. If you want to modify only some data, you must configure a second route map section and specify an explicit match.

---

**Examples**

This example shows that the routes with their next hop out Ethernet interface 0 is distributed:

```
fwsM/context_name(config)# route-map name
fwsM(config-route-map)# match interface ethernet 0
```

---

**Related Commands**

- ip local pool
- match (route map submode)
- match interface (route map submode)
- match ip next-hop (route map submode)
- match ip route-source (route map submode)
- match metric (route map submode)
- route-map
- set metric (route map submode)
- set metric-type (route map submode)

## match ip next-hop (route map submode)

To redistribute any routes that have a next-hop router address that is passed by one of the access lists specified, use the **match ip next-hop** command in the route-map submode. To remove the next-hop entry, use the **no** form of this command.

```
[no] match ip next-hop {acl-id...}
```

Syntax Description	<i>acl-id</i>	Number of a standard or extended access list; valid values are from 1 to 199.
--------------------	---------------	---

Defaults	Routes are distributed freely, without being required to match a next-hop address.
----------	--

Command Modes	Security Context Mode: single context mode Access Location: system and context command line Command Mode: configuration mode Firewall Mode: routed firewall mode
---------------	---

Command History	Release	Modification
	1.1(1)	Support for this command was introduced on the FWSM.

Usage Guidelines	An ellipsis (...) in the command syntax indicates that your command input can include multiple values for the access-list-number or access-list-name argument.
------------------	--

The **route-map global** configuration command and the **match** and **set route-map** configuration commands allow you to define the conditions for redistributing routes from one routing protocol into another. Each **route-map** command has a **match** and **set** commands that are associated with it. The **match** commands specify the match criteria—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular redistribution actions to perform if the criteria that is 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 give the **match** commands in any order. All **match** commands must “pass” to cause the route to be redistributed according to the set actions given with the **set** commands. The **no** forms of the **match** commands remove the specified match criteria.

When you are passing 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 and specify an explicit match.

---

**Examples**

This example shows how to distribute routes that have a next-hop router address passed by access list 5 or 80:

```
fws(config)# route-map name
fws(config-route-map)# match ip next-hop 5 80
```

---

**Related Commands**

**ip local pool**  
**match (route map submode)**  
**match interface (route map submode)**  
**match ip route-source (route map submode)**  
**match metric (route map submode)**  
**route-map**  
**set metric (route map submode)**  
**set metric-type (route map submode)**

## match ip route-source (route map submode)

To redistribute routes that have been advertised by routers and access servers at the address that is specified by the access lists, use the **match ip route-source** command in the route-map submode. To remove the route-source entry, use the **no** form of this command.

```
[no] match ip route-source {acl-id ...}
```

### Syntax Description

<i>acl-id</i>	Number of a standard or extended access list; valid values are from 1 to 199.
---------------	---

### Defaults

No filtering on a route source.

### Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

An ellipsis (...) in the command syntax indicates that your command input can include multiple values for the access-list-number or access-list-name argument.

The **route-map global** configuration command and the **match** and **set route-map** configuration commands allow you to 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—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular 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 give the **match** commands in any order. All **match** commands must “pass” to cause the route to be redistributed according to the set actions given with the **set** commands. The **no** forms of the **match** commands remove the specified match criteria.

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 and specify an explicit match. The next-hop and source-router address of the route are not the same in some situations.

---

**Examples**

This example shows how to distribute routes that have been advertised by routers and access servers at the addresses specified by access lists 5 and 80:

```
fws(config)# route-map name
fws(config-route-map)# match ip route-source 5 80
```

---

**Related Commands**

- ip local pool**
- match (route map submode)**
- match interface (route map submode)**
- match ip next-hop (route map submode)**
- match metric (route map submode)**
- route-map**
- set metric (route map submode)**
- set metric-type (route map submode)**

## match metric (route map submode)

To redistribute routes with the metric specified, use the **match metric** command in the route-map submode. To remove the entry, use the **no** form of this command.

[no] **match metric** *number*

<b>Syntax Description</b>	<i>number</i>	Route metric, which can be an IGRP five-part metric; valid values are from 0 to 4294967295.
---------------------------	---------------	---

**Defaults** No filtering on a metric value.

**Command Modes**

- Security Context Mode: single context mode
- Access Location: system and context command line
- Command Mode: configuration mode
- Firewall Mode: routed firewall mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	1.1(1)	Support for this command was introduced on the FWSM.

**Usage Guidelines**

The **route-map global** configuration command and the **match** and **set route-map** configuration commands allow you to 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—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular redistribution actions to perform if the criteria that is 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. The **match** commands can be given 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** forms of the **match** commands remove the specified match criteria.

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 and specify an explicit match.

**Examples** This example shows how to redistribute routes with the metric 5:

```
fws(config)# route-map name
fws(config-route-map)# match metric 5
```

**Related Commands**

**ip local pool**  
**match (route map submode)**  
**match interface (route map submode)**  
**match ip next-hop (route map submode)**  
**match ip route-source (route map submode)**  
**route-map**  
**set metric (route map submode)**  
**set metric-type (route map submode)**

## match route-type (route map submode)

To redistribute routes of the specified type, use the **match route-type** command in the route-map submode. To remove the route type entry, use the **no** form of this command.

```
[no] match route-type {local | internal | {external [type-1 | type-2]} | nssa-external | [type-1 | type-2]}
```

Syntax Description		
<b>local</b>		Locally generated Border Gateway Protocol (BGP) routes.
<b>internal</b>		Open Shortest Path First (OSPF) intra-area and interarea routes or Enhanced Interior Gateway Routing Protocol (EIGRP) internal routes.
<b>external</b>		OSPF external routes or EIGRP external routes.
<b>type-1</b>		(Optional) Specifies the route type 1.
<b>type-2</b>		(Optional) Specifies the route type 2.
<b>nssa-external</b>		Specifies the external not-so-stubby-area (NSSA).

**Defaults** This command is disabled by default.

**Command Modes**

- Security Context Mode: single context mode
- Access Location: system and context command line
- Command Mode: configuration mode
- Firewall Mode: routed firewall mode

Command History	Release	Modification
	1.1(1)	Support for this command was introduced on the FWSM.

**Usage Guidelines**

The **route-map global** configuration command and the **match** and **set route-map** configuration commands allow you to 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—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular redistribution actions to perform if the criteria that is 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 give the **match** commands in any order. All **match** commands must “pass” to cause the route to be redistributed according to the set actions given with the **set** commands. The **no** forms of the **match** commands remove the specified match criteria.

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 and specify an explicit match.

For OSPF, the **external type-1** keywords match only type 1 external routes and the **external type-2** keywords match only type 2 external routes.

---

**Examples**

This example shows how to redistribute internal routes:

```
fws(config)# route-map name
fws(config-route-map)# match route-type internal
```

---

**Related Commands**

**ip local pool**  
**match (route map submode)**  
**match interface (route map submode)**  
**match ip next-hop (route map submode)**  
**match metric (route map submode)**  
**route-map**  
**set metric (route map submode)**  
**set metric-type (route map submode)**

## member (context submode)

To determine the class to which a context belongs, use the **member** command in the context submode. To remove a context from a class, use the **no** form of this command.

**member** *class\_name*

[no] **member** *class\_name*

### Syntax Description

<i>class_name</i>	Specifies a class name.
-------------------	-------------------------

### Command Modes

Security Context Mode: multiple context mode

Access Location: system command line

Command Mode: privileged mode

Firewall Mode: routed firewall mode and transparent firewall mode

### Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

The class sets resource limitations for each class member. To use the settings of a class, assign the context to the class. All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to a default using this command. See the **class** command to add a class. You can assign a context to one resource class only. An exception is that limits that are undefined in the member class are inherited from the default class; a context could be a member of a default plus another class.

### Examples

This example shows how to assign a context to a class:

```
fwsm/context_name(config)# member regulus
```

### Related Commands

Other Context Subconfiguration Commands

**allocate-interface (context submode)**

**config-url (context submode)**

**limit-resource**

Other Related Commands

**admin-context**

**changeto**

**class**

**clear context**

**show class**

**show context**

# mgcp

To configure additional support for the Media Gateway Control Protocol (MGCP) fixup (packet application inspection), use the **mgcp** command. To remove MGCP support, use the **no** form of this command.

[no] **mgcp call-agent** *ip\_address group\_id*

[no] **mgcp command-queue** *limit*

[no] **mgcp gateway** *ip\_address group\_id*

## Syntax Description

<b>call-agent</b> <i>ip_address</i>	Specifies the IP address of the call agent.
<b>command-queue</b> <i>limit</i>	Specifies the maximum number of commands to queue; valid values are from 1 to 4294967295.
<b>gateway</b> <i>ip_address</i>	Specifies the IP address of the gateway.
<i>group_id</i>	Call agent group ID; valid values are from 0 to 4294967295.

## Defaults

The MGCP command queue *limit* is 200.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **mgcp** command allows you to provide additional support for the MGCP fixup. The MGCP fixup is enabled with the **fixup protocol mgcp** command.

The **mgcp call-agent** command is used to specify a group of call agents that can manage one or more gateways. The call agent group information is used to open connections for the call agents in the group (other than the one to which the gateway sends a command) so that any of the call agents can send the response. Call agents with the same *group\_id* belong to the same group. A call agent may belong to more than one group.

The **mgcp command-queue** command allows you to specify the maximum number of MGCP commands that are queued while waiting for a response. When the limit has been reached and a new command arrives, the command that has been in the queue for the longest time is removed.

The **mgcp gateway** command allows you to specify which group of call agents are managing a particular gateway. The IP address of the gateway is specified with the *ip\_address* argument. The *group\_id* argument must correspond with the *group\_id* of the call agents that are managing the gateway. A gateway may belong to one group only.

---

**Examples**

This example shows how to limit the MGCP command queue to 150 commands, allows call agents 10.10.11.5 and 10.10.11.6 to control gateway 10.10.10.115, and allows call agents 10.10.11.7 and 10.10.11.8 to control both gateways 10.10.10.116 and 10.10.10.117:

```
fws(config)# mgcp call-agent 10.10.11.5 101
fws(config)# mgcp call-agent 10.10.11.6 101
fws(config)# mgcp call-agent 10.10.11.7 102
fws(config)# mgcp call-agent 10.10.11.8 102
fws(config)# mgcp command-queue 150
fws(config)# mgcp gateway 10.10.10.115 101
fws(config)# mgcp gateway 10.10.10.116 102
fws(config)# mgcp gateway 10.10.10.117 102
```

---

**Related Commands**

- clear mgcp**
- debug**
- fixup protocol**
- show conn**
- show mgcp**
- timeout**

# mkdir

To create a new directory, use the **mkdir** command.

```
mkdir [disk:] path
```

## Syntax Description

<b>disk:</b>	(Optional) Changes the current working directory.
<i>path</i>	Specifies the path for the new directory.

## Defaults

If you do not specify a directory, the directory is changed to **disk:**.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: system command line  
 Command Mode: privileged mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

If a directory with the same name already exists, then the new directory is not created. The **mkdir disk:** command prompts you to enter a directory name.

## Examples

This example shows how to make a new directory:

```
fws(config)# mkdir disk:
Create directory filename [running-config]? my_context.cfg
Created dir disk:my_context.cfg
fws(config)# dir
Directory of disk:/
 1   -rw-  1519      10:03:50 Jul 14 2003   my_context1.cfg
 2   -rw-  1516      10:04:02 Jul 14 2003   my_context2.cfg
 3   -rw-  1516      10:01:34 Jul 14 2003   admin.cfg
10   drw-    0         09:24:38 Jul 16 2003   my_context3.cfg
60985344 bytes total (60968960 bytes free)
```

**Related Commands**

cd  
copy disk  
copy flash  
copy startup-config  
copy tftp  
dir  
format  
more  
pwd  
rename  
rmdir  
show file

# mode

To change the FWSM to single context mode or multiple context mode, use the **mode** command.

```
mode { single | multiple }
```

Syntax Description	single	Sets the FWSM to the single context mode.
	multiple	Sets the FWSM to the multiple context mode.

**Defaults** The default setting depends on whether Cisco shipped the FWSM to you with the Security Context feature enabled (multiple context mode), or whether you are upgrading your FWSM (single context mode).

**Command Modes** Security Context Mode: single context mode and multiple context mode  
 Access Location: System and Context  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

Command History	Release	Modification
	2.2(1)	Support for this command was introduced on the FWSM.

**Usage Guidelines** This command allows you to change the behavior of the FWSM and prompts you to reboot the module. By default, multiple mode allows you to use two contexts. To enable more than two contexts, you must enter an activation key (if it was not already entered by Cisco).

If you are changing from single context mode to multiple context mode, the FWSM converts the running configuration into two files: a new startup.cfg (in the Flash) that has the system configuration and admin.cfg (in the disk partition) that has the admin context. The original running configuration is saved as old\_running.cfg (in disk). The original startup configuration is not saved.

If you convert from multiple context mode to single context mode, the startup configuration is not automatically converted back to the original running configuration. You must copy the backup version to the running and startup configurations. Because the system configuration does not have any network interfaces as part of its configuration, you must session into the FWSM from the switch to perform the copy as follows:

```
fwsM# copy disk:old_running.cfg running-config
fwsM# copy running-config startup-config
```

**mode**

---

**Examples**

This example shows how to change the context mode:

```
fwsm# mode multiple
```

---

**Related Commands**

**show mode**

# monitor-interface

To enable interface monitoring on a specific interface within a context, use the **monitor-interface** command.

**[no] monitor-interface** *interface\_name*

## Syntax Description

*interface\_name* Specifies the name of the interface being monitored.

## Defaults

Not configured

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The number of interfaces that can be monitored for the FWSM is 250 per module. Hello messages are exchanged during every interface poll frequency time period between the FWSM failover pair. The failover interface poll time is 3 to 15 seconds. For example, if the poll time is set to 5 seconds, testing begins on an interface if 5 consecutive hellos are not heard on that interface (25 seconds).

Monitored failover interfaces can have the following status:

- Unknown—Initial status. This status can also mean the status cannot be determined.
- Normal—The interface is receiving traffic.
- Testing—Hello messages are not heard on the interface for five poll times.
- Link Down—The VLAN for the interface is shut down.
- No Link—VLANs for the interface are not configured.
- Failed—No traffic is received on the interface, yet traffic is heard on the peer interface.

## Examples

This example shows how to start interface monitoring:

```
fwsn(config)# monitor-interface inside
```

**Related Commands**

**clear failover**  
**failover**  
**failover interface ip address**  
**failover interface-policy**  
**failover lan interface**  
**failover lan unit**  
**failover link**  
**failover polltime**  
**failover replication http**  
**failover reset**  
**show failover**  
**show monitor-interface**  
**write standby**

## more

To display the contents of a file, use the **more** command.

**more** [/ascii] || [/binary] [disk:] *path*

Syntax Description		
/ascii	(Optional)	Displays a binary file in binary mode and an ASCII file in binary mode.
/binary	(Optional)	Displays any file in binary mode.
disk:	(Optional)	Changes the current working directory.
<i>path</i>		Specifies the path for the new directory.

**Defaults** ACSII mode

**Command Modes** Security Context Mode: single context mode and multiple context mode  
 Access Location: system command line  
 Command Mode: privileged mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

Command History	Release	Modification
	2.2(1)	Support for this command was introduced on the FWSM.

**Usage Guidelines** The **more disk:** command prompts you to enter a filename.

**Examples** This example shows how to display the contents of a file named “test.cfg”:

```
fwsd(config)# more test.cfg
: Saved
: Written by enable_15 at 10:04:01 Jul 14 2003

FWSM Version 2.2(0)141
nameif vlan300 outside security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname test
fixup protocol ftp 21
fixup protocol h323 H225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
```

```

access-list deny-flow-max 4096
access-list alert-interval 300
access-list 100 extended permit icmp any any
access-list 100 extended permit ip any any
pager lines 24
icmp permit any outside
mtu outside 1500
ip address outside 172.29.145.35 255.255.0.0
no pdm history enable
arp timeout 14400
access-group 100 in interface outside
!
interface outside
!
route outside 0.0.0.0 0.0.0.0 172.29.145.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02 rpc 0:10:00 h3
23 0:05:00 h225 1:00:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
snmp-server host outside 128.107.128.179
snmp-server location my_context, USA
snmp-server contact admin@my_context.com
snmp-server community public
no snmp-server enable traps
floodguard enable
fragment size 200 outside
no sysopt route dnat
telnet timeout 5
ssh timeout 5
terminal width 511
gdb enable
mgcp command-queue 0
Cryptochecksum:0000000000000000000000000000000000000000
: end

```

**Related Commands**

```

cd
copy disk
copy flash
copy startup-config
copy tftp
dir
format
mkdir
pwd
rename
rmdir
show file

```

# mtu

To specify the maximum transmission unit (MTU) for an interface, use the **mtu** command. To reset the MTU block size to 1500 for Ethernet interfaces, use the **no** form of this command.

**[no] mtu** *interface\_name* *bytes*

## Syntax Description

<i>interface_name</i>	Internal or external network interface name.
<i>bytes</i>	Number of bytes in the MTU; valid values are from 64 to 65,535 bytes.

## Defaults

*bytes* is 1500 for Ethernet interfaces.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **mtu** command allows you to set the data size that is sent on a connection. Data that is larger than the MTU value is fragmented before being sent.

The FWSM supports IP path MTU discovery (as defined in RFC 1191), which allows a host to dynamically discover and cope with the differences in the maximum allowable MTU size of the various links along the path. Sometimes, the FWSM cannot forward a datagram because the packet is larger than the MTU that you set for the interface, but the “don’t fragment” (DF) bit is set. The network software sends a message to the sending host, alerting it to the problem. The host has to fragment packets for the destination so that they fit the smallest packet size of all the links along the path.

The default MTU is 1500 bytes in a block for Ethernet interfaces (which is also the maximum). This value is sufficient for most applications, but you can pick a lower number if network conditions require it.

When using the Layer 2 Timeline Protocol (L2TP), we recommend that you set the MTU size to 1380 to account for the L2TP header and IPsec header length.

## Examples

This example shows how to specify the MTU for an interface:

```
fwsM/context_name(config)# mtu inside 8192
fwsM/context_name(config)# show mtu
fwsM/context_name(config)# mtu outside 1500
fwsM/context_name(config)# mtu inside 8192
```

■ mtu

---

**Related Commands**    show mtu

# name

To associate a name with an IP address, use the **name** command. To enable the association, use the **names** command. To disable the use of the text names but not remove them from the configuration, use the **no** form of this command.

**[no] name ip\_address name**

**names**

## Syntax Description

<i>ip_address</i>	IP address of the host that is named.
<i>name</i>	Name assigned to the IP address.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

Use the **names** command to enable association of a name with an IP address.

When defining the *name*, you can use characters a to z, A to Z, 0 to 9, a dash, and an underscore. The *name* cannot start with a number. If the name is over 16 characters, the **name** command fails.

The **name** command allows you to identify a host by a text name and map text strings to IP addresses. The **no names** command allows you to disable the use of the text names but does not remove them from the configuration. Use the **clear name** command to clear the list of names from the FWSM configuration.

You must first use the **names** command before you use the **name** command. Use the **name** command immediately after you use the **names** command and before you use the **write memory** command.

To disable displaying **name** values, use the **no names** command.

You can associate only one name with an IP address.

Both the **name** and **names** commands are saved in the configuration.

While the **name** command lets you assign a name to a network mask, no other FWSM command requiring a mask lets you use the name as a mask value. For example, this command is accepted:

```
fwsm/context_name(config)# name 255.255.255.0 class-C-mask
```

**Note**

None of the commands in which a mask is required can process the “class-C-mask” as an accepted network mask.

**Examples**

This example shows that the **names** command allows you to enable use of the **name** command. The **name** command substitutes **fwsd\_inside** for references to 192.168.42.3 and **fwsd\_outside** for 209.165.201.3. You can use these names with the **ip address** commands when assigning IP addresses to the network interfaces. The **no names** command disables the **name** command values from displaying. Subsequent use of the **names** command again restores the **name** command value display.

```
fwsd(config)# names
fwsd(config)# name 192.168.42.3 fwsd_inside
fwsd(config)# name 209.165.201.3 fwsd_outside
fwsd(config)# ip address inside fwsd_inside 255.255.255.0
fwsd(config)# ip address outside fwsd_outside 255.255.255.224
```

```
fwsd(config)# show ip address
System IP Addresses:
  inside ip address fwsd_inside mask 255.255.255.0
  outside ip address fwsd_outside mask 255.255.255.224
```

```
fwsd(config)# no names
fwsd(config)# show ip address
System IP Addresses:
  inside ip address 192.168.42.3 mask 255.255.255.0
  outside ip address 209.165.201.3 mask 255.255.255.224
```

```
fwsd(config)# names
fwsd(config)# show ip address
System IP Addresses:
  inside ip address fwsd_inside mask 255.255.255.0
  outside ip address fwsd_outside mask 255.255.255.224
```

**Related Commands**

**clear name**  
**show name**

# nameif

To name interfaces and assign the security level, use the **nameif** command.

```
no nameif interface interface_name security_lvl

no nameif interface [interface_name] [security_lvl]
```

## Syntax Description

<i>interface</i>	VLAN name or mapped name.
<i>interface_name</i>	Name for the network interface; this name can have up to 48 characters.
<i>security_lvl</i>	Specifies the security level; valid values are from 0–100.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

You cannot change the name of an interface; you can only change the security level. The interface identification is either *vlan num*, or for multiple mode, it is the mapped name that is configured with the **allocate interface** command. There is no hardware ID for the FWSM, only VLAN IDs are allowed.



### Caution

If you enter the **no nameif** command, all configurations that use that name are removed.

The security level between two interfaces determines the way the adaptive security algorithm is applied. A lower *security\_level* interface is outside a higher level interface, and equivalent interfaces are outside each other. Refer to the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Software Configuration Guide* for more information about security levels.

## Examples

This example shows a configuration in single mode:

```
fws(config)# nameif vlan18 perimeter1 sec50
fws(config)# nameif vlan23 perimeter2 sec20
```

This example shows a configuration in multiple mode:

```
fws(config-context)# allocate-interface vlan7 intf-out
fws(config-context)# allocate-interface vlan17 intf-in
```

```
fws(config-context)# allocate-interface vlan23 intf-dmz
fws(config-context)# changeto context_name
fws/context_name(config)# nameif intf-out outside security0
fws/context_name(config)# nameif intf-in inside security90
fws/context_name(config)# nameif intf-dmz dmz security50
```

---

**Related Commands**    **allocate-interface (context submode)****interface****global****nat****static**

# names

To enable IP address to the name conversions that you can configure with the **name** command, use the **names** command. To disable address to name conversion, use the **no** form of this command.

**[no] names**

## Syntax Description

This command has no arguments or keywords.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Examples

This example shows how to enable names:

```
fws(config)# names
```

## Related Commands

**clear name**  
**name**  
**show name**  
**show names**

# nat

To associate a network with a pool of global IP addresses, use the **nat** command. To remove the **nat** command, use the **no** form of this command.

```
[no] nat local_interface nat_id local_ip [mask [dns] [outside] [[tcp] tcp_max_conns [emb_limit]
[norandomseq]]] [udp udp_max_conns]
```

```
[no] nat local_interface nat_id access-list access_list_name [dns] [outside] [[tcp] tcp_max_conns
[emb_limit] [norandomseq]]] [udp udp_max_conns]
```

## Syntax Description

<i>local_interface</i>	Name of the network interface as specified by the <b>nameif</b> command through which the hosts or network that are designated by <i>local_ip</i> are accessed.
<i>nat_id</i>	ID of the group of host or networks; see the “Usage Guidelines” section for valid values.
<i>local_ip</i>	Internal network IP address to be translated.
<i>mask</i>	(Optional) IP netmask to apply to the <i>local_ip</i> .
<b>dns</b>	(Optional) Specifies to use the created translation to rewrite the DNS address record.
<b>outside</b>	(Optional) Specifies that the <b>nat</b> command apply to the outside interface address.
<b>norandomseq</b>	(Optional) Disables TCP Initial Sequence Number (ISN) randomization protection.
<b>tcp</b>	(Optional) Specifies that the maximum TCP connections and embryonic limit are set for the TCP protocol.
<i>tcp_max_conns</i>	(Optional) Maximum number of simultaneous connections that the <i>local_ip</i> hosts allow. Idle connections are closed after the time that is specified by the <b>timeout connection</b> command.
<i>emb_limit</i>	(Optional) Maximum number of embryonic connections per host.
<b>udp</b>	(Optional) Specifies a maximum number of UDP connection parameters that can be configured.
<i>udp_max_conns</i>	(Optional) Sets the maximum number of simultaneous UDP connections that the <i>local_ip</i> hosts are each allowed to use. Idle connections are closed after the time that is specified by the <b>timeout connection</b> command.
<b>access-list</b> <i>access_list_name</i>	Specifies the traffic to exempt from Network Address Translation (NAT) processing, based on the access list that is specified by <i>access_list_name</i> .
<b>access-list</b> <i>access_list_name</i>	(Optional) Associates <b>access-list</b> commands to the <b>nat 0</b> command and exempts traffic that matches the specified access list from NAT processing.

## Defaults

The defaults are as follows:

- *emb\_limit* is 0.
- **udp** is not required.

<b>Command Modes</b>	Security Context Mode: single context mode and multiple context mode Access Location: context command line Command Mode: configuration mode Firewall Mode: routed firewall mode
----------------------	--

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	1.1(1)	Support for this command was introduced on the FWSM.
	2.2(1)	This command was modified to support UDP maximum connections for local hosts.

<b>Usage Guidelines</b>	<p>An embryonic connection is a connection request that has not finished the necessary handshake between source and destination.</p> <p>The <b>nat</b> command allows you to enable or disable address translation for one or more internal addresses. Address translation occurs when a host starts an outbound connection and the IP addresses in the internal network are translated into global addresses. NAT allows your network to have any IP addressing scheme and the FWSM protects these addresses from visibility on the external network.</p>
-------------------------	--



<b>Note</b>	The number of address translations allowed is per each FWSM. The FWSM supports 2,048 address translations for the <b>nat</b> command, 1,051 address translations for the <b>global</b> command, and 2,048 address translations for the <b>static</b> command. The FWSM also supports up to 4,096 access control entries (ACEs) in ACLs used for policy NAT.
-------------	---



<b>Note</b>	The FWSM does not support NAT with a Cisco CallManager inside the firewall with IP phones outside the firewall because NAT does not support TFTP messages.
-------------	--

The **outside** keyword lets you enable or disable address translation for the external addresses. For access control, IPSec, and AAA, use the real outside address.



<b>Note</b>	Enabling outside Port Address Translation (PAT) can make the FWSM vulnerable to a flood DoS attack. We recommend that you restrict the address range specified with the <b>nat nat_id local_ip mask outside</b> command. In addition, you should set the connection limit to a value that accounts for the memory capacity of the FWSM. A PAT session is made up of a PAT xlate and an UDP or TCP connection. A PAT xlate consumes about 120 bytes and a TCP or UDP connection consumes 250 bytes.
-------------	--

The **nat interface\_name 0 access-list access\_list\_name** command allows you to exempt traffic that is matched by the **access-list** commands from the NAT services. The extent to which the inside hosts are accessible from the outside depends on the **access-list** commands that you use to permit inbound access. The *interface\_name* is the higher security level interface name. The *access\_list\_name* is the name that you use to identify the **access-list** command.

Adding the **access-list** keyword changes the behavior of the **nat 0** command. Without the **access-list** keyword, the command is backward compatible with previous versions. The **nat 0** command disables NAT. Specifically, proxy ARPing for the IP addresses is disabled when you enter the **nat 0** command.

**Note**

The access list that you specify with the **nat 0 access-list** command does not work with an **access-list** command that contains a port specification. The following sample commands will not work:

```
fwsM/context_name(config)# access-list no-nat permit tcp host xx.xx.xx.xx host YY.YY.YY.YY
fwsM/context_name(config)# nat (inside) 0 access-list no-nat
```

After changing or removing the **nat** command, use the **clear xlate** command.

The connection limit lets you set the maximum number of outbound connections that can be started with the IP address criteria that you specify. This limit lets you prevent a type of attack where processes are started without being completed.

Use the **no nat** command to remove the **nat** command.

See [Table 2-10](#) for a list of interface access commands. The security levels for the demilitarized zones are 40 for dmz1 and 60 for dmz2.

**Table 2-10 Interface Access Commands by Interface**

From This Interface	To This Interface	Use This Command	From This Interface	To This Interface	Use This Command
inside	outside	<b>nat</b>	dmz2	outside	<b>nat</b>
inside	dmz1	<b>nat</b>	dmz2	dmz1	<b>nat</b>
inside	dmz2	<b>nat</b>	dmz2	inside	<b>static</b>
dmz1	outside	<b>nat</b>	outside	dmz1	<b>static</b>
dmz1	dmz2	<b>static</b>	outside	dmz2	<b>static</b>
dmz1	inside	<b>static</b>	outside	inside	<b>static</b>

To obtain access from a higher security level interface to a lower security level interface, use the **nat** command. From a lower security level interface to a higher security level interface, use the **static** command.

Enable identity address translation with the **nat 0** command. The **nat 0** command requires that traffic initiates from an inside host. Use this command when you have IP addresses that are the same as those commands that are used on more than one interface. The extent to which the inside hosts are accessible from the outside depends on the **access-list** commands that permit inbound access.

Addresses on each interface must be on a different subnet.

Entering the **nat 0 10.2.3.0** command allows those IP addresses in the 10.2.3.0 net to appear on the outside without translation. All other hosts are translated depending on how their **nat** commands appear in the configuration.

Entering the **nat 1 0 0** command allows all outbound connections to pass through the FWSM with address translation. If you use the **nat (inside) 1 0 0** command, you can start connections on any interface with a lower security level on both the perimeter interfaces and the outside interface. With NAT, you must also use the **global** keyword to provide a pool of addresses through which translated connections pass. The NAT ID must be the same on the **nat** and **global** commands.

Entering the **nat 1 10.2.3.0** command allows only outbound connections originating from the inside host 10.2.3.0 to pass through the FWSM to go to their destinations with address translation.

When specifying the network mask for *local\_ip*, you can use 0.0.0.0 to allow all outbound connections to translate with IP addresses from the global pool. The netmask 0.0.0.0 can be abbreviated as 0.

The *nat\_id* is referenced by the **global** command to associate a global pool with the *local\_ip*.

*nat\_id* values can be 0, **0 access list** *access\_list\_name*, or any number from 1 to 256. A *nat\_id* of 0 indicates that no address translation takes place for *local\_ip*.

A *nat\_id* of **0 access list** *access\_list\_name* specifies the traffic to exempt from NAT processing, based on the access list that is specified by the *access\_list\_name*. This command is useful in a VPN configuration where traffic between private networks should be exempted from NAT.

A *nat\_id* that is a number from 1 to 256 specifies the inside hosts for dynamic address translation. The dynamic addresses are chosen from a global address pool that is created when you enter the **global** command. The *nat\_id* number must match the *global\_id* number of the global address pool that you want to use for dynamic address translation.

The *local\_ip* determines the group of hosts or networks that are referred to by *nat\_id*. You can use 0.0.0.0 to allow all hosts to start outbound connections. The 0.0.0.0 *local\_ip* can be abbreviated as 0. An IP address not found in a more explicit *nat\_id* group defaults to a less explicit or a **0** which indicates the least explicit.

Idle connections are closed after the idle timeout is specified by the **timeout conn** command.

In both the **nat** and **static** statements, the *udp\_max\_conn* field is applicable even when the TCP *max\_conns* limit is not set, by using the keyword **udp**. This allows the two limits to be exclusively configured.

## Examples

This example shows how to make the addresses visible from the outside network:

```
fwsM/context_name(config)# nat (inside) 0 209.165.201.0 255.255.255.224
fwsM/context_name(config)# static (inside, outside) 209.165.201.0 209.165.201.0 netmask
255.255.255.224
fwsM/context_name(config)# access-list acl_out permit host 10.0.0.1 209.165.201.0
255.255.255.224 eq ftp
fwsM/context_name(config)# access-group acl_out in interface outside

fwsM/context_name(config)# nat (inside) 0 209.165.202.128 255.255.255.224
fwsM/context_name(config)# static (inside, outside) 209.165.202.128 209.165.202.128
netmask 255.255.255.224
fwsM/context_name(config)# access-list acl_out permit tcp host 10.0.0.1 209.165.202.128
255.255.255.224 eq ftp
fwsM/context_name(config)# access-group acl_out in interface outside
...
```

This example shows how to use the **nat 0 access-list** command to permit access to internal host 10.1.1.15 through the inside interface, “inside,” to bypass NAT when connecting to outside host 10.2.1.3:

```
fwsM/context_name(config)# access-list no-nat permit ip host 10.1.1.15 host 10.2.1.3
fwsM/context_name(config)# nat (inside) 0 access-list no-nat
```

This command shows how to disable all NAT on the FWSM with three interfaces:

```
fwsM/context_name(config)# access-list all-ip-packet permit ip 0 0 0 0
fwsM/context_name(config)# nat (dmz) 0 access-list all-ip-packet
fwsM/context_name(config)# nat (inside) 0 access-list all-ip-packet
```

These examples show how to specify that all the hosts on the 10.0.0.0 and 3.3.3.0 inside networks can start outbound connections:

```
fwsM/context_name(config)# nat (inside) 1 10.0.0.0 255.0.0.0
fwsM/context_name(config)# global (outside) 1 209.165.201.25-209.165.201.27 netmask
255.255.255.224
fwsM/context_name(config)# global (outside) 1 209.165.201.30

fwsM/context_name(config)# nat (inside) 3 10.3.3.0 255.255.255.0
fwsM/context_name(config)# global (outside) 3 209.165.201.10-209.165.201.25 netmask
255.255.255.224
```

### Related Commands

- access-list deny-flow-max
- clear nat
- global
- interface
- nameif
- show nat
- static

# object-group

To define object groups that you can use to optimize your configuration, use the **object-group** command. Use the **no** form of this command to remove object groups from the configuration.

**[no] object-group icmp-type** *obj\_grp\_id*

icmp-type group subcommands  
**description** *description\_text*  
**icmp-object** *icmp\_type*

**[no] object-group network** *obj\_grp\_id*

network group subcommands  
**description** *description\_text*  
**network-object** **host** *host\_addr* | *host\_name*  
**network-object** *net\_addr* *netmask*  
**group-object**

**[no] object-group protocol** *obj\_grp\_id*

protocol group subcommands  
**description** *description\_text*  
**protocol-object** *protocol*

**[no] object-group service** *obj\_grp\_id* {**tcp** | **udp** | **tcp-udp**}

service group subcommands  
**description** *description\_text*  
**port-object** **range** *begin\_service* *end\_service*  
**port-object** **eq** *service*

## Syntax Description

<b>icmp-type</b>	Defines a group of ICMP types such as echo and echo-reply. After entering the main <b>object-group icmp-type</b> command, add ICMP objects to the ICMP type group with the <b>icmp-object</b> and the <b>group-object</b> subcommand.
<i>obj_grp_id</i>	Identifies the object group (one to 64 characters) and can be any combination of letters, digits, and the “_”, “-”, “.” characters.
<b>description</b> <i>description_text</i>	Adds a description of up to 200 characters to an object-group.
<b>icmp-object</b>	Adds ICMP objects to an ICMP-type object group.
<i>icmp_type</i>	Decimal number or name of an ICMP type.
<b>network</b>	Defines a group of hosts or subnet IP addresses. After entering the main <b>object-group network</b> command, add network objects to the network group with the <b>network-object</b> and the <b>group-object</b> subcommand.
<b>network-object</b>	Adds network objects to a network object group.
<b>host</b>	Defines a host object.
<i>host_addr</i>	Host IP address or host name (if the host name is already defined using the <b>name</b> command).
<i>host_name</i>	Host name (if the host name is not defined using the <b>name</b> command).

<i>net_addr</i>	Network address; used with <i>netmask</i> to define a subnet object.
<i>netmask</i>	Netmask; used with <i>net_addr</i> to define a subnet object.
<b>group-object</b>	Adds network object groups.
<b>protocol</b>	Defines a group of protocols such as TCP and UDP. After entering the main <b>object-group protocol</b> command, add protocol objects to the protocol group with the <b>protocol-object</b> and the <b>group-object</b> subcommand.
<b>protocol-object</b>	Adds protocol objects to a protocol object group.
<i>protocol</i>	Protocol name or number.
<b>service</b>	Defines a group of TCP/UDP port specifications such as “eq smtp” and “range 2000 2010.” After entering the main <b>object-group service</b> command, add port objects to the service group with the <b>port-object</b> and the <b>group-object</b> subcommand.
<b>tcp</b>	Specifies that service group is used for TCP.
<b>udp</b>	Specifies that service group is used for UDP.
<b>tcp-udp</b>	Specifies that service group can be used for TCP and UDP.
<b>port-object</b>	<b>object-group service</b> subcommand used to add port objects to a service object group.
<b>range</b>	Specifies the range parameters.
<i>begin_service</i>	Specifies the decimal number or name of a TCP or UDP port that is the beginning value for a range of services.
<i>end_service</i>	Specifies the decimal number or name of a TCP or UDP port that is the ending value for a range of services.
<b>eq service</b>	Specifies the decimal number or name of a TCP or UDP port for a service object.

**Command Modes**

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

**Command History**

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

**Usage Guidelines**

Objects such as hosts, protocols, or services can be grouped, and then you can issue a single command using the group name to apply to every item in the group.

When you define a group with the **object-group** command and then use any FWSM command, the command applies to every item in that group. This feature can significantly reduce your configuration size.

Once you define an object group, you must use the **object-group** keyword before the group name in all applicable FWSM commands as follows:

```
fwsms# show object-group group_name
```

where *group\_name* is the name of the group.

This example shows the use of an object group once it is defined:

```
fwsM/context_name (config) # access-list access_list_name permit tcp any object-group
group_name
```

In addition, you can group the **access-list** command arguments as shown in [Table 2-11](#).

**Table 2-11 Individual Arguments and Object Group Replacements**

Individual Arguments	Object Group Replacement
<i>protocol</i>	<b>object-group</b> <i>protocol</i>
<i>host and subnet</i>	<b>object-group</b> <i>network</i>
<i>service</i>	<b>object-group</b> <i>service</i>
<i>icmp_type</i>	<b>object-group</b> <i>icmp_type</i>

You can group commands hierarchically; an object group can be a member of another object group.

To use object groups, you must do the following:

- Use the **object-group** keyword before the object group name in all commands as follows:

```
fwsM/context_name (config) # access-list acl permit tcp object-group remotes
object-group locals object-group eng_svc
```

where *remotes* and *locals* are sample object group names.

- The object group must be nonempty.
- You cannot remove or empty an object group if it is currently being used in a command.

After you enter a main **object-group** command, the command mode changes to its corresponding submenu. The object group is defined in the submenu. The active mode is indicated in the command prompt format. For example, the prompt in the configuration terminal mode appears as follows:

```
fwsM_name (config-type) #
```

where *fwsM\_name* is the name of the FWSM.

However, when you enter the **object-group** command, the prompt appears as follows:

```
fwsM#_name (config-type) #
```

where *fwsM\_name* is the name of the FWSM, and *type* is the object-group type.

Use the **exit**, **quit**, or any valid config-mode commands such as **access-list** to close an **object-group** submenu and exit the **object-group** main command.

The **show object-group** command displays all defined object groups by their *grp\_id* when the **show object-group id grp\_id** command is entered, and by their group type when you enter the **show object-group grp\_type** command. When you enter the **show object-group** command without an argument, all defined object groups are shown.

Use the **no object-group** command to remove a group of previously defined **object-group** commands. Without an argument, the **clear object-group** command allows you to remove all defined object groups that are not being used in a command. The *grp\_type* argument removes all defined object groups that are not being used in a command for that group type only.

See [Table 2-12](#) for a listing of ICMP type numbers and names.

**Table 2-12 ICMP Types**

Number	ICMP Type Name
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	address-mask-request
18	address-mask-reply
31	conversion-error
32	mobile-redirect

You can use all other FWSM commands in submode, including the **show** and **clear** commands.

Subcommands appear indented when displayed or saved by the **show config**, **write**, or **config** commands.

Subcommands have the same command privilege level as the main command.

When you use more than one object group in an **access-list** command, the elements of all object groups that are used in the command are linked together, starting with the first group's elements with the second group's elements, then the first and second group's elements together with the third group's elements, and so on.

The starting position of the description text is the character right after the white space (a blank or a tab) following the **description** keyword.

## Examples

This example shows how to use the **object-group icmp-type** submode to create a new icmp-type object group:

```
fws(config)# object-group icmp-type icmp-allowed
fws(config-icmp-type)# icmp-object echo
fws(config-icmp-type)# icmp-object time-exceeded
fws(config-icmp-type)# exit
```

This example shows how to use the **object-group network** subcommand to create a new network object group:

```
fwsd(config)# object-group network sjc_eng_ftp_servers
fwsd(config-network)# network-object host sjc.eng.ftp.servcers
fwsd(config-network)# network-object host 172.23.56.194
fwsd(config-network)# network-object 192.1.1.0 255.255.255.224
fwsd(config-network)# exit
```

This example shows how to use the **object-group network** subcommand to create a new network object group and map it to an existing object-group:

```
fwsd(config)# object-group network sjc_ftp_servers
fwsd(config-network)# network-object host sjc.ftp.servers
fwsd(config-network)# network-object host 172.23.56.195
fwsd(config-network)# network-object 193.1.1.0 255.255.255.224
fwsd(config-network)# group-object sjc_eng_ftp_servers
fwsd(config-network)# exit
```

This example shows how to use the **object-group protocol** submode to create a new protocol object group:

```
fwsd(config)# object-group protocol proto_grp_1
fwsd(config-protocol)# protocol-object udp
fwsd(config-protocol)# protocol-object ipsec
fwsd(config-protocol)# exit
```

```
fwsd(config)# object-group protocol proto_grp_2
fwsd(config-protocol)# protocol-object tcp
fwsd(config-protocol)# group-object proto_grp_1
fwsd(config-protocol)# exit
```

This example shows how to use the **object-group service** submode to create a new port (service) object group:

```
fwsd(config)# object-group service eng_service tcp
fwsd(config-service)# group-object eng_www_service
fwsd(config-service)# port-object eq ftp
fwsd(config-service)# port-object range 2000 2005
fwsd(config-service)# exit
```

This example shows how to add and remove a text description to an object group:

```
fwsd(config)# object-group protocol protos1
fwsd(config-protocol)# description This group of protocols is for our internal network

fwsd(config-protocol)# show object-group id protos1
object-group protocol protos1
description: This group of protocols is for our internal network

fwsmdocipsecl(config-protocol)# no description
fwsmdocipsecl(config-protocol)# show object-group id protos1
object-group protocol protos1
```

This example shows how to use the **group-object** submode to create a new object group that consists of previously defined objects:

```
fwsd(config)# object-group network host_grp_1
fwsd(config-network)# network-object host 192.168.1.1
fwsd(config-network)# network-object host 192.168.1.2
fwsd(config-network)# exit

fwsd(config)# object-group network host_grp_2
fwsd(config-network)# network-object host 172.23.56.1
```

```
fwsd(config-network)# network-object host 172.23.56.2
fwsd(config-network)# exit

fwsd(config)# object-group network all_hosts
fwsd(config-network)# group-object host_grp_1
fwsd(config-network)# group-object host_grp_2
fwsd(config-network)# exit

fwsd(config)# access-list grp_1 permit tcp object-group host_grp_1 any eq ftp
fwsd(config)# access-list grp_2 permit tcp object-group host_grp_2 any eq smtp
fwsd(config)# access-list all permit tcp object-group all_hosts any eq www
```

Without the **group-object** command, you need to define the *all\_hosts* group to include all the IP addresses that have already been defined in *host\_grp\_1* and *host\_grp\_2*. With the **group-object** command, the duplicated definitions of the hosts are eliminated.

These examples show how to use object groups to simplify the access list configuration:

```
fwsd/context_name(config)# object-group network remote
fwsd/context_name(config-network)# network-object host kqk.suu.dri.ixx
fwsd/context_name(config-network)# network-object host kqk.suu.py1.gnl

fwsd/context_name(config)# object-group network locals
fwsd/context_name(config-network)# network-object host 172.23.56.10
fwsd/context_name(config-network)# network-object host 172.23.56.20
fwsd/context_name(config-network)# network-object host 172.23.56.194
fwsd/context_name(config-network)# network-object host 172.23.56.195

fwsd/context_name(config)# object-group service eng_svc ftp
fwsd/context_name(config-service)# port-object eq www
fwsd/context_name(config-service)# port-object eq smtp
fwsd/context_name(config-service)# port-object range 25000 25100
```

This grouping enables the access list to be configured in 1 line instead of 24 lines, which would be needed if no grouping is used. Instead, with the grouping, the access list configuration is as follows:

```
fwsd/context_name(config)# access-list acl permit tcp object-group remote object-group
locals object-group eng_svc
```



#### Note

The **show config** and **write** commands allow you to display the access list as configured with the object group names. The **show access-list** command displays the access list entries that are expanded out into individual entries without their object groupings.

#### Related Commands

```
clear object-group
show object-group
```

## ospf (interface submode)

To configure interface-specific Open Shortest Path First (OSPF) parameters, use the **ospf** command in the interface submode. To return to the default setting, use the **no** form of this command.

```
ospf { authentication [message-digest | null] } | { authentication-key password } | { cost
  interface_cost } | { database-filter all out } | { dead-interval seconds } | { hello-interval
  seconds } | { message-digest-key key-id md5 key } | { mtu-ignore } | { priority number } |
  { retransmit-interval seconds } | { transmit-delay seconds }
```

```
no ospf
```

### Syntax Description

<b>authentication</b>	Specifies the authentication type for an interface.
<b>message-digest</b>	(Optional) Specifies to use OSPF message digest authentication.
<b>null</b>	(Optional) Specifies to not use OSPF authentication.
<b>authentication-key password</b>	Assigns an OSPF authentication password for use by neighboring routing devices.
<b>cost interface_cost</b>	Specifies the cost (a link-state metric) of sending a packet through an interface; valid values are from 0 to 255, expressed as the link-state metric.
<b>database-filter all out</b>	Filters out outgoing link-state advertisements (LSAs) to an OSPF interface.
<b>dead-interval seconds</b>	Sets the interval before declaring that a neighboring routing device is down if no hello packets are received; valid values are from 1 to 65535 seconds.
<b>hello-interval seconds</b>	Specifies the interval between hello packets that are sent on the interface; valid values are from 1 to 65535 seconds.
<b>message-digest-key key_id</b>	Enables the Message Digest 5 (MD5) authentication and specifies the numerical authentication key ID number; valid values are from 1 to 255.
<b>md5 key</b>	Alphanumeric password of up to 16 bytes.
<b>mtu-ignore</b>	Disables OSPF maximum transmission unit (MTU) mismatch detection on receiving database packets.
<b>priority number</b>	Specifies the priority of the router; valid values are from 0 to 255.
<b>retransmit-interval seconds</b>	Specifies the time between LSA retransmissions for adjacent routers belonging to the interface; valid values are from 1 to 65535 seconds.
<b>transmit-delay seconds</b>	Sets the estimated time that is required to send a link-state update packet on the interface; valid values are from 1 to 65535 seconds.

### Defaults

The defaults are as follows:

- OSPF routing is disabled on the FWSM interfaces.
- **mtu-ignore** is enabled.
- **authentication** is **null** (no area authentication).
- **dead-interval** is four times the interval set by the **ospf hello-interval** command.

- **hello-interval** *seconds* is 10 seconds.
- **retransmit-interval** *seconds* is 5 seconds.
- **transmit-delay** *seconds* is 1 second.

### Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

The **routing interface** command is the main command for all interface-specific OSPF interface mode commands. Enter this command with the name of the FWSM interface (*interface\_name*) that you want to configure, and then proceed with interface-specific configuration through the **routing interface** subcommands.

Once you enter the **routing interface** command, the command prompt appears as (config-routing)#, indicating that you are in the submode.

The **show routing interface** command allows you to display the configuration for the interface specified.

#### ospf authentication

The **ospf authentication** [**message-digest** | **null**] subcommand allows you to specify the authentication type for an interface. To remove the authentication type for an interface, use the **no ospf authentication** [**message-digest** | **null**] subcommand. The default for authentication is **null**, which means that there is no authentication. The **null** subcommand overrides password or message digest authentication (if configured) for an OSPF area.

#### ospf authentication-key

The **ospf authentication-key** *password* subcommand allows you to assign a password to be used by neighboring routers that are using the OSPF simple password authentication. The *password* argument can be any continuous string of characters that can be entered from the keyboard up to 8 bytes.

The **no ospf authentication-key** subcommand allows you to remove a previously assigned OSPF password.

#### ospf cost

The **ospf cost** *interface\_cost* subcommand allows you to explicitly specify the cost of sending a packet on an interface. The *interface\_cost* parameter is an unsigned integer value from 0 to 255.

The **no ospf cost** subcommand allows you to reset the path cost to the default value.

**ospf database-filter all out**

The **ospf database-filter** subcommand allows you to filter outgoing link-state advertisements (LSAs) to an OSPF interface. The **no ospf database-filter all out** subcommand allows you to restore the forwarding of LSAs to the interface.

**ospf dead-interval**

The **ospf dead-interval** *seconds* subcommand allows you to set the dead interval before neighbors to declare the router down (the length of time during which no hello packets are seen). *seconds* specifies the dead interval and must be the same for all nodes on the network. The default for *seconds* is four times the interval set by the **ospf hello-interval** command from 1 to 65535. The **no ospf dead-interval** subcommand allows you to return to the default interval value.

**ospf hello-interval**

The **ospf hello-interval** *seconds* subcommand allows you to specify the interval between hello packets that the FWSM sends on the interface. The **no ospf hello-interval** subcommand allows you to return to the default interval. The default is 10 seconds with a range from 1 to 65535.

**ospf mtu-ignore**

The **ospf mtu-ignore** subcommand allows you to disable OSPF MTU mismatch detection on receiving DBD packets and is enabled by default.

**ospf message-digest-key** *key\_id md5 key*

The **ospf message-digest-key** *key\_id md5 key* subcommand allows you to enable OSPF Message Digest 5 (MD5) authentication. The **no ospf message-digest-key** *key\_id md5 key* subcommand allows you to remove an old MD5 key. *key\_id* is a numerical identifier from 1 to 255 for the authentication key. *key* is an alphanumeric password of up to 16 bytes. White space characters, such as a tab or space, are not supported. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

**ospf priority**

The **ospf priority** *number* subcommand allows you to set the router priority, which helps determine the designated router for this network. The **no ospf priority** *number* subcommand allows you to return to the default value.

**ospf retransmit-interval**

The **ospf retransmit-interval** *seconds* subcommand allows you to specify the time between LSA retransmissions for adjacencies belonging to the interface. The **no ospf retransmit-interval** subcommand allows you to return to the default value. The default value is 5 seconds with a range from 1 to 65535.

**ospf transmit-delay**

The **ospf transmit-delay** *seconds* subcommand allows you to set the estimated time required to send a link-state update packet on the interface. The **no ospf transmit-delay** subcommand allows you to return to the default value. The default value is 1 second with a range from 1 to 65535.

---

**Examples**

This example shows how to enter the submode on the outside interface of the FWSM (needed to configure OSPF routing):

```
fwsm(config)# routing interface outside
```

In the routing submode, the command prompt appears as “(config-routing)#.”

This example shows the configuration for two concurrently running OSPF processes, with the IDs 5 and 12, on the outside interface of the firewall:

```
fws(config)# routing interface
fws(config)# show ospf

Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x      0
Number of opaque AS LSA 0. Checksum Sum 0x      0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 0. 0 normal 0 stub 0 nssa
External flood list length 0

Routing Process "ospf 12" with ID 172.23.59.232 and Domain ID 0.0.0.12
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x      0
Number of opaque AS LSA 0. Checksum Sum 0x      0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 0. 0 normal 0 stub 0 nssa
External flood list length 0
```

This example shows how to change the retransmit interval to 15 seconds:

```
fws(config-interface)# ospf retransmit-interval 15
```

### Related Commands

```
clear ospf
routing interface
show ospf
show ospf border-routers
show ospf database
show ospf flood-list
show ospf interface
show ospf neighbor
show ospf request-list
show ospf retransmission-list
show ospf summary-address
show ospf virtual-links
show routing interface
```

# pager

To enable screen paging, use the **pager** command. To disable screen paging and let the output display without interruption, use the **no** form of this command.

**[no] pager [lines lines]**

## Syntax Description

**lines lines** (Optional) Specifies the number of lines before the “---more---” prompt appears; valid values are from 1 to 25.

## Defaults

*number* is 25.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system and context command line

Command Mode: Unprivileged

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

If you set the **pager lines** command to a value and want to revert back to the default, enter the **pager** command without keywords or arguments. This command is session based. If the pager value is changed in a session, the value is not changed globally for other sessions. Use the **pager 0** command to disable paging.

When you enable paging, the “---more---” prompt appears. The “---more---” prompt uses syntax that is similar to the UNIX **more** command as follows:

- To display another screenful, press the **Space** bar.
- To display the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

## Examples

This example shows how to enable screen paging:

```
fws(config)# pager lines 2
fws(config)# ping inside 10.0.0.42
    10.0.0.42 NO response received -- 1010ms
    10.0.0.42 NO response received -- 1000ms
<--- more --->
```

## Related Commands

**clear pager**  
**show pager**

# password/passwd

To set the password for Telnet access to the FWSM console, use the **password** command.

```
{password | passwd} password [encrypted]
```

## Syntax Description

<i>password</i>	Case-sensitive password of up to 16 alphanumeric and special characters.
<b>encrypted</b>	(Optional) Specifies that the password you entered is already encrypted.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: system and context command line  
 Command Mode: privileged mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **password/passwd** command allows you to set a password for Telnet access to the FWSM console. The **passwd** keyword is also accepted as a shortened form of **password**. Additionally, the FWSM configuration displays the password using the short form, **passwd**.

Any character can be used in the password except a question mark and a space. The *password* that you specify with the **encrypted** keyword must be 16 characters.

An empty password is changed into an encrypted string. However, any use of the **write** command displays or writes the passwords in encrypted form. Once passwords are encrypted, they are not reversible back to plain text.



### Note

Write down the new password and store it in a manner consistent with your site's security policy. Once you change this password, you cannot see it again.

## Examples

This example shows how to set the password for Telnet access to the FWSM console:

```
fwsM(config)# (config)# password watag00s1am
fwsM(config)# show password
passwd jMorNbK0514fadBh encrypted
```

**Related Commands**

clear passwd  
enable  
show passwd  
telnet

# pdm

To configure the support communication between the FWSM and a browser running the PDM, use the **pdm** command.

**pdm disconnect** *session\_id*

[no] **pdm history enable**

**pdm history** [view {**all** | **12h** | **5d** | **60m** | **10m**}] [snapshot] [feature {**all** | **blocks** | **cpu** | **failover** | **ids** | **interface** *interface\_name* | **memory** | **perfmon** | **xlates**}] [**pdmclient**]

**pdm group** *real\_group\_name* *associated\_intf\_name*

**pdm group** *ref\_group\_name* *ref\_intf\_name* **reference** *real\_group\_name*

**pdm location** *ip\_address* *netmask* *interface\_name*

**pdm logging** [*level* [*messages*]]

## Syntax Description

<b>disconnect</b> <i>session_id</i>	Disconnects the specified PDM session from the FWSM.
<b>history enable</b>	Enables PDM data sampling.
<b>view</b> <i>type</i>	(Optional) Specifies the PDM history view to display; valid values for the type argument are 12 hours ( <b>12h</b> ), 5 days ( <b>5d</b> ), 60 minutes ( <b>60m</b> ), 10 minutes ( <b>10m</b> ), or <b>all</b> history contents in the PDM history buffer.
<b>snapshot</b>	(Optional) Displays only the last PDM history data point.
<b>feature</b>	(Optional) Specifies to display the history for a single feature.
<b>all</b>	(Optional) Displays the history for all the features.
<b>blocks</b>	(Optional) Displays the blocks used for the feature.
<b>cpu</b>	(Optional) Displays the history for CPU usage.
<b>failover</b>	(Optional) Displays the history for failover.
<b>ids</b>	(Optional) Displays the history for the Intrusion Detection System.
<b>interface</b> <i>interface_name</i>	(Optional) Specifies the interface name on which the PDM resides.
<b>memory</b>	(Optional) Displays the history for memory; similar to output of the <b>show memory</b> command.
<b>perfmon</b>	(Optional) Displays the history for performance.
<b>xlates</b>	(Optional) Displays the history for translation slot information.
<b>pdmclient</b>	(Optional) Displays the PDM history in PDM-display format.
<i>real_group_name</i>	Name of a PDM object group that contains real IP addresses.
<i>associated_intf_name</i>	Name of the interface to which the specified object group is associated.
<i>ref_group_name</i>	Name of an object group that contains network address-translated IP addresses of the object group specified by <i>real_group_name</i> .
<i>ref_intf_name</i>	Name of the interface from which the destination IP address of inbound traffic is network address translated.
<b>reference</b>	Associates an object group that contains real IP addresses to an object group that contains NAT IP addresses.

<i>ip_address</i>	Specifies the host or network on which the PDM resides.
<i>netmask</i>	Specifies the network mask for the <b>pdm location</b> <i>ip_address</i> .
<b>location</b>	Associates an interface with an IP address on which PDM resides.
<b>logging</b>	Specifies the type and number of syslog messages that are displayed through the PDM <b>syslog</b> keyword.
<i>level</i>	(Optional) Specifies the priority level of syslog messages that are displayed in the PDM <b>syslog</b> keyword.
<i>messages</i>	(Optional) Specifies the maximum number of messages that are stored in the PDM buffer before the buffer discards the old messages.

### Defaults

The defaults are as follows:

- The PDM syslog *level* is 0.
- The logging *messages* is 100.
- The maximum is 512.

### Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system and context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.
2.2(1)	The PDM software version 2.2 was used to configure the FWSM release 1.1(1). In this release, the PDM has been replaced by the Firewall Device Manager (FDM).

### Usage Guidelines

The *associated\_intf\_name* name is defined by the **nameif** command.

The *ref\_intf\_name* name is defined by the **nameif** command.

The **pdm location** command is an internal PDM command.

Once the message buffer exceeds the specified *message*, old messages are discarded.

The **pdm history enable** command allows you to enable the PDM data sampling. If not specified, the history for all features is displayed. PDM data sampling takes a data sample and stores the sample data to the PDM history buffer. The **no** form of this command disables PDM data sampling.

The **pdm disconnect** command and the **show pdm sessions** commands are accessible through the FWSM command-line interface.

The **failover** keyword history display is similar to the output of the **show failover** command.

The **memory** keyword history display is similar to the output of the **show perfmon** command.

The **xlates** keyword history display is similar to the output of the **show xlate** command.

The **clear pdm**, **pdm group**, **pdm history**, **pdm location**, and **pdm logging** commands may appear in your configuration, but they are designed to work as internal PDM-to-FWSM commands that are accessible only to the PDM.

You can only associate one interface to an *ip\_address/netmask* pair when you enter the **pdm location** command. Specifying a new pair replaces the old definition.

## Examples

This example shows how to report the last data point in PDM-display format:

```
fwsd(config)# pdm history enable
fwsd(config)# show pdm history view 10m snapshot pdmclient
INTERFACE|outside|up|IBC|0|OBC|1088|IPC|0|OPC|0|IBR|17|OBR|0|IPR|0|OPR|0|IERR|1|NB|0|RB|0|
RNT|0|GNT|0|CRC|0|FRM|0|OR|0|UR|0|OERR|0|COLL|0|LCOLL|0|RST|0|DEF|0|LCR|0:FWSMoutsideINTER
FACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|1952|METRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTOR
Y|SNAP|IPR|VIEW|10|17|METRIC_HISTORY|SNAP|OPR|VIEW|10|1|METRIC_HISTORY|SNAP|IERR|VIEW|10|0|
|METRIC_HISTORY|SNAP|OERR|VIEW|10|0|:FWSMinsideINTERFACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|0|
|METRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTORY|SNAP|IPR|VIEW|10|0|METRIC_HISTORY|SNAP|
OPR|VIEW|10|1|METRIC_HISTORY|SNAP|IERR|VIEW|10|0|METRIC_HISTORY|SNAP|OERR|VIEW|10|0|:FWSMS
YS:METRIC_HISTORY|SNAP|MEM|VIEW|10|52662272|METRIC_HISTORY|SNAP|BLK4|VIEW|10|1600|METRIC_H
ISTORY|SNAP|BLK80|VIEW|10|400|METRIC_HISTORY|SNAP|BLK256|VIEW|10|998|METRIC_HISTORY|SNAP|B
LK1550|VIEW|10|676|METRIC_HISTORY|SNAP|XLATES|VIEW|10|0|METRIC_HISTORY|SNAP|CONNS|VIEW|10|
0|METRIC_HISTORY|SNAP|TCPCONNS|VIEW|10|0|METRIC_HISTORY|SNAP|UDPCONNS|VIEW|10|0|METRIC_HIS
TORY|SNAP|URLS|VIEW|10|0|METRIC_HISTORY|SNAP|WEBSNS|VIEW|10|0|METRIC_HISTORY|SNAP|TCPFIXUP
S|VIEW|10|0|METRIC_HISTORY|SNAP|TCPINTERCEPTS|VIEW|10|0|METRIC_HISTORY|SNAP|HTTPFIXUPS|VIE
W|10|0|METRIC_HISTORY|SNAP|FTPFIXUPS|VIEW|10|0|METRIC_HISTORY|SNAP|AAAAUTHENUPS|VIEW|10|0|
METRIC_HISTORY|SNAP|AAAAUTHORUPS|VIEW|10|0|METRIC_HISTORY|SNAP|AAAACCOUNTS|VIEW|10|0|
```

This example shows how to report the data formatted for the FWSM CLI:

```
fwsd(config)# pdm history enable
fwsd(config)# show pdm history view 10m snapshot
Available 4 byte Blocks: [ 10s] : 1600
Used 4 byte Blocks: [ 10s] : 0
Available 80 byte Blocks: [ 10s] : 400
Used 80 byte Blocks: [ 10s] : 0
Available 256 byte Blocks: [ 10s] : 500
Used 256 byte Blocks: [ 10s] : 0
Available 1550 byte Blocks: [ 10s] : 931
Used 1550 byte Blocks: [ 10s] : 385
Available 1552 byte Blocks: [ 10s] : 0
Used 1552 byte Blocks: [ 10s] : 0
Available 2560 byte Blocks: [ 10s] : 0
Used 2560 byte Blocks: [ 10s] : 0
Available 4096 byte Blocks: [ 10s] : 0
Used 4096 byte Blocks: [ 10s] : 0
Available 8192 byte Blocks: [ 10s] : 0
Used 8192 byte Blocks: [ 10s] : 0
Available 16384 byte Blocks: [ 10s] : 0
Used 16384 byte Blocks: [ 10s] : 0
Available 65536 byte Blocks: [ 10s] : 0
Used 65536 byte Blocks: [ 10s] : 0
CPU Utilization: [ 10s] : 0
IP Options Bad: [ 10s] : 0
Record Packet Route: [ 10s] : 0
IP Options Timestamp: [ 10s] : 0
Provide s,c,h,tcc: [ 10s] : 0
Loose Source Route: [ 10s] : 0
SATNET ID: [ 10s] : 0
Strict Source Route: [ 10s] : 0
IP Fragment Attack: [ 10s] : 0
Impossible IP Attack: [ 10s] : 0
IP Teardrop: [ 10s] : 0
```

```
ICMP Echo Reply: [ 10s] : 0
ICMP Unreachable: [ 10s] : 0
ICMP Source Quench: [ 10s] : 0
ICMP Redirect: [ 10s] : 0
ICMP Echo Request: [ 10s] : 0
ICMP Time Exceeded: [ 10s] : 0
ICMP Parameter Problem: [ 10s] : 0
ICMP Time Request: [ 10s] : 0
ICMP Time Reply: [ 10s] : 0
ICMP Info Request: [ 10s] : 0
ICMP Info Reply: [ 10s] : 0
ICMP Mask Request: [ 10s] : 0
ICMP Mask Reply: [ 10s] : 0
Fragmented ICMP: [ 10s] : 0
Large ICMP: [ 10s] : 0
Ping of Death: [ 10s] : 0
No Flags: [ 10s] : 0
SYN & FIN Only: [ 10s] : 0
FIN Only: [ 10s] : 0
FTP Improper Address: [ 10s] : 0
FTP Improper Port: [ 10s] : 0
Bomb: [ 10s] : 0
Snork: [ 10s] : 0
Chargen: [ 10s] : 0
DNS Host Info: [ 10s] : 0
DNS Zone Transfer: [ 10s] : 0
DNS Zone Transfer High Port: [ 10s] : 0
DNS All Records: [ 10s] : 0
Port Registration: [ 10s] : 0
Port Unregistration: [ 10s] : 0
RPC Dump: [ 10s] : 0
Proxied RPC: [ 10s] : 0
ypserv Portmap Request: [ 10s] : 0
ypbind Portmap Request: [ 10s] : 0
yppasswd Portmap Request: [ 10s] : 0
ypupdated Portmap Request: [ 10s] : 0
ypxfrd Portmap Request: [ 10s] : 0
mountd Portmap Request: [ 10s] : 0
rexcd Portmap Request: [ 10s] : 0
rexcd Attempt: [ 10s] : 0
statd Buffer Overflow: [ 10s] : 0
Input KByte Count: [ 10s] : 41804
Output KByte Count: [ 10s] : 526456
Input KPacket Count: [ 10s] : 364
Output KPacket Count: [ 10s] : 450
Input Bit Rate: [ 10s] : 0
Output Bit Rate: [ 10s] : 0
Input Packet Rate: [ 10s] : 0
Output Packet Rate: [ 10s] : 0
Input Error Packet Count: [ 10s] : 0
No Buffer: [ 10s] : 0
Received Broadcasts: [ 10s] : 90076
Runts: [ 10s] : 0
Giants: [ 10s] : 0
CRC: [ 10s] : 0
Frames: [ 10s] : 0
Overruns: [ 10s] : 0
Underruns: [ 10s] : 0
Output Error Packet Count: [ 10s] : 0
Collisions: [ 10s] : 8895
LCOLL: [ 10s] : 0
Reset: [ 10s] : 0
Deferred: [ 10s] : 3138
Lost Carrier: [ 10s] : 0
```

```

Hardware Input Queue: [ 10s] : 128
Software Input Queue: [ 10s] : 0
Hardware Output Queue: [ 10s] : 0
Software Output Queue: [ 10s] : 0
Input KByte Count: [ 10s] : 61835
Output KByte Count: [ 10s] : 26722
Input KPacket Count: [ 10s] : 442
Output KPacket Count: [ 10s] : 418
Input Bit Rate: [ 10s] : 0
Output Bit Rate: [ 10s] : 0
Input Packet Rate: [ 10s] : 0
Output Packet Rate: [ 10s] : 0
Input Error Packet Count: [ 10s] : 0
No Buffer: [ 10s] : 0
Received Broadcasts: [ 10s] : 308607
Runts: [ 10s] : 0
Giants: [ 10s] : 0
CRC: [ 10s] : 0
Frames: [ 10s] : 0
Overruns: [ 10s] : 0
Underruns: [ 10s] : 0
Output Error Packet Count: [ 10s] : 0
Collisions: [ 10s] : 0
LCOLL: [ 10s] : 0
Reset: [ 10s] : 0
Deferred: [ 10s] : 2
Lost Carrier: [ 10s] : 707
Hardware Input Queue: [ 10s] : 128
Software Input Queue: [ 10s] : 0
Hardware Output Queue: [ 10s] : 0
Software Output Queue: [ 10s] : 0
Available Memory: [ 10s] : 45293568
Used Memory: [ 10s] : 21815296
Xlate Count: [ 10s] : 0
Connection Count: [ 10s] : 0
TCP Connection Count: [ 10s] : 0
UDP Connection Count: [ 10s] : 0
URL Filtering Count: [ 10s] : 0
URL Server Filtering Count: [ 10s] : 0
TCP Fixup Count: [ 10s] : 0
TCP Intercept Count: [ 10s] : 0
HTTP Fixup Count: [ 10s] : 0
FTP Fixup Count: [ 10s] : 0
AAA Authentication Count: [ 10s] : 0
AAA Authorzation Count: [ 10s] : 0
AAA Accounting Count: [ 10s] : 0
Current Xlates: [ 10s] : 0
Max Xlates: [ 10s] : 0
ISAKMP SAs: [ 10s] : 0
IPSec SAs: [ 10s] : 0
L2TP Sessions: [ 10s] : 0
L2TP Tunnels: [ 10s] : 0
PPTP Sessions: [ 10s] : 0
PPTP Tunnels: [ 10s] : 0

```

**Related Commands**

```

clear pdm
fixup protocol
setup
show pdm

```

# perfmon

To display performance information, use the **perfmon** command.

**perfmon** { **verbose** | **interval** *seconds* | **quiet** | **settings** }

## Syntax Description

<b>verbose</b>	Displays performance monitor information at the FWSM console.
<b>interval</b> <i>seconds</i>	Specifies the number of seconds before the performance display is refreshed on the console.
<b>quiet</b>	Disables the performance monitor displays.
<b>settings</b>	Displays the interval and whether it is quiet or verbose.

## Defaults

The *seconds* is 120 seconds.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: privileged mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **perfmon** command allows you to monitor the performance of the FWSM. Use the **show perfmon** command to display the information immediately. Use the **perfmon verbose** command to display the information every 2 minutes continuously. Use the **perfmon interval** *seconds* command with the **perfmon verbose** command to display the information continuously every number of seconds that you specify.

An example of the performance information is displayed as follows:

PERFMON STATS:	Current	Average
Xlates	33/s	20/s
Connections	110/s	10/s
TCP Conns	50/s	42/s
WebSns Req	4/s	2/s
TCP Fixup	20/s	15/s
HTTP Fixup	5/s	5/s
FTP Fixup	7/s	4/s
AAA Authen	10/s	5/s

AAA Author	9/s	5/s
AAA Account	3/s	3/s

This information lists the number of translations, connections, Websense requests, address translations (called “fixups”), and AAA transactions that occur each second.

### Examples

This example shows how to display the performance monitor statistics every 30 seconds on the FWSM console:

```
fwsm/context_name(config)# perfmon interval 120  
fwsm/context_name(config)# perfmon quiet  
fwsm/context_name(config)# perfmon settings  
interval: 120 (seconds)  
quiet
```

### Related Commands

**show perfmon**

# ping

To determine if other IP addresses are visible from the FWSM, use the **ping** command.

```
ping [interface_name] ip_address
```

## Syntax Description

<i>interface_name</i>	(Optional) Internal or external network interface name.
<i>ip_address</i>	IP address of a host on the inside or outside networks.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: system and context command line  
 Command Mode: privileged mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **ping** command allows you to determine if the FWSM has connectivity or if a host is available on the network. If the FWSM is connected also ensure that the **icmppermit any interface** command is configured. This configuration is required to allow the FWSM to respond and accept these messages. The command output shows if the response was received. If a host is not responding, when you enter the **ping** command, you see the display “NO response received.” Use the **show interface** command to ensure that the FWSM is connected to the network and is passing traffic.

The address of the specified *interface\_name* is used as the source address of the ping.

If you want internal hosts to ping external hosts, you must create an ICMP **access-list** command for an echo reply; for example, to give ping access to all hosts, use the **access-list acl\_grp permit icmp any any** command and bind the **access-list** command to the interface that you want to test using the **access-group** command.

If you are pinging through the FWSM between hosts or routers, but the pings are not successful, use the **debug icmp trace** command to monitor the success of the ping. Pings are successful when they are both inbound and outbound.

The FWSM **ping** command does not require an interface name. If you do not specify an interface name, the FWSM checks the routing table to find the address that you specify. You can specify an interface name to indicate through which interface the ICMP echo requests are sent.

An example of the usage is as follows:

```
fwsd(config)# ping 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
```

```
10.0.0.1 response received -- 0ms
```

You can enter the command specifying the interface as follows:

```
fws(config)# ping outside 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 0ms
```

---

**Examples**

This example shows how to determine if other IP addresses are visible from the FWSM:

```
fws(config)# ping 192.168.42.54
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
```

---

**Related Commands**

**icmp**

**show interface**

# privilege

To configure the command privilege levels, use the **privilege** command. To disallow the configuration, use the **no** form of this command.

```
[no] privilege [show | clear | configure] level level [mode {enable | configure}] command
command
```

Syntax Description		
<b>show</b>	(Optional) Sets the privilege level for the <b>show</b> command corresponding to the command specified.	
<b>clear</b>	(Optional) Sets the privilege level for the <b>clear</b> command corresponding to the command specified.	
<b>configure</b>	(Optional) Sets the privilege level for the <b>configure</b> command corresponding to the command specified.	
<b>level</b> <i>level</i>	Specifies the privilege level; valid values are from 0 to 15.	
<b>mode enable</b>	(Optional) Indicates that the level is for the enable mode of the command.	
<b>mode configure</b>	(Optional) Indicates that the level is for the configure mode of the command.	
<b>command</b> <i>command</i>	Specifies the command on which to set the privilege level.	

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **privilege** command allows you to set user-defined privilege levels for the FWSM commands. This command is useful for setting different privilege levels for related configuration, **show**, and **clear** commands. Make sure that you verify privilege level changes in your commands with your security policies before using the new privilege levels.

When commands and users have privilege levels set, the two are compared to determine if a given user can execute a given command. If the user's privilege level is lower than the privilege level of the command, the user is prevented from executing the command.

To change between privilege levels, use the **login** command to access another privilege level and the appropriate **logout**, **exit**, or **quit** command to exit that level.

The **mode enable** and **mode configure** keywords are for commands with both enable and configure modes.

Lower privilege level numbers are lower privilege levels.

**Note**

The **aaa authentication** and **aaa authorization** commands need to include any new privilege levels that you define before you can use them in your AAA server configuration.

**Examples**

This example shows how to set the privilege level “5” for an individual user as follows:

```
username intern1 password pass1 privilege 5
```

This example shows how to define a set of **show** commands with the privilege level “5” as follows:

```
fws(config)# level:

fws(config)# privilege show level 5 command alias
fws(config)# privilege show level 5 command apply
fws(config)# privilege show level 5 command arp
fws(config)# privilege show level 5 command auth-prompt
fws(config)# privilege show level 5 command blocks
```

This example shows how to apply privilege level 11 to a complete AAA authorization configuration:

```
fws(config)# privilege configure level 11 command aaa
fws(config)# privilege configure level 11 command aaa-server
fws(config)# privilege configure level 11 command access-group
fws(config)# privilege configure level 11 command access-list
fws(config)# privilege configure level 11 command activation-key
fws(config)# privilege configure level 11 command age
fws(config)# privilege configure level 11 command alias
fws(config)# privilege configure level 11 command apply
```

**Related Commands**

**aaa authentication**  
**clear privilege**  
**login**  
**object-group**  
**show curpriv**  
**show privilege**  
**username**

# pwd

To display the current working directory, use the **pwd** command.

**pwd**

## Syntax Description

This command has no arguments or keywords.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system command line

Command Mode: privileged mode

Firewall Mode: Routed and Transparent

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Examples

This example shows how to display the current working directory:

```
fwsd(config)# pwd
disk:
```

## Related Commands

**cd**  
**copy disk**  
**copy flash**  
**copy startup-config**  
**copy tftp**  
**dir**  
**format**  
**mkdir**  
**more**  
**rename**  
**rmdir**  
**show file**

# quit

To exit the current privilege level or mode, use the **quit** command.

**quit**

---

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

---

**Defaults** This command has no default settings.

---

**Command Modes** Security Context Mode: single context mode and multiple context mode  
 Access Location: System Context Command Line  
 Command Mode: Unprivileged  
 Firewall Mode: routed firewall mode and transparent firewall mode

---

Command History	Release	Modification
	1.1(1)	Support for this command was introduced on the FWSM.

---



---

**Usage Guidelines** You may also use the key sequence **^Z** to exit.

---

**Examples** This example shows how to use the **quit** command:

```
fws(config)# quit
fws# quit
fws>
```

---

**Related Commands** **exit**

# redistribute

To configure redistribution between the Open Shortest Path First (OSPF) processes according to the specified parameters, use the **redistribute** command. To remove redistribution configurations, use the **no** form of this command.

```
redistribute {static | connected} [metric metric_value] [metric-type metric_type] [route-map
map_name] [tag tag_value] [subnets]
```

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

## Syntax Description

<b>static</b>	Specifies the static interface.
<b>connected</b>	Specifies the connected interface.
<b>metric</b> <i>metric_value</i>	(Optional) Specifies the OSPF default metric value from 0 to 16777214.
<b>metric-type</b> <i>metric_type</i>	(Optional) Specifies the OSPF metric type; valid values are <b>type-1</b> , <b>type-2</b> , <b>internal</b> , or <b>external</b> .
<b>route-map</b> <i>map_name</i>	(Optional) Name of the route map to apply.
<b>tag</b> <i>tag_value</i>	(Optional) Specifies the value to match for controlling redistribution with route maps.
<b>subnets</b>	(Optional) Specifies for redistributing routes into OSPF and scopes the redistribution for the specified protocol.
<b>ospf</b> <i>pid</i>	Specifies an internally used identification parameter for an OSPF routing process; valid values are from 1 to 65535.
<b>match</b>	(Optional) Specifies the conditions for redistributing routes from one routing protocol into another.
<b>internal</b> <i>type</i>	Specifies OSPF metric routes that are internal to a specified autonomous system; valid values are <b>1</b> or <b>2</b> .
<b>external</b> <i>type</i>	Specifies the OSPF metric routes that are external to a specified autonomous system; valid values are <b>1</b> or <b>2</b> .
<b>nssa-external</b> <i>type</i>	Specifies the OSPF metric type for routes that are external to a not-so-stubby area (NSSA); valid values are <b>1</b> or <b>2</b> .

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode

Access Location: system command line

Command Mode: configuration mode

Firewall Mode: Routed

## redistribute

### Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

### Usage Guidelines

The **show ospf** command allows you to display the configured **router ospf** subcommands.

You assign the *pid* locally on the FWSM; it can be from 1 to 65535. You must assign a unique value for each OSPF routing process.

### Examples

This example shows how to configure redistribution between the OSPF processes according to the specified parameters:

```
fws(config)# redistribute static
```

### Related Commands

```
router ospf
show ip ospf
show redistribute
```

# reload

To reboot and reload the configuration, use the **reload** command..

**reload [noconfirm]**

Syntax Description	noconfirm	(Optional) Permits the FWSM to reload without user confirmation.
--------------------	-----------	--

Defaults	This command has no default settings.
----------	---------------------------------------

Command Modes	Security Context Mode: single context mode and multiple context mode Access Location: systemcommand line Command Mode: privileged mode Firewall Mode: routed firewall mode and transparent firewall mode
---------------	---

Command History	Release	Modification
	1.1(1)	Support for this command was introduced on the FWSM.

Usage Guidelines	<p>The <b>reload</b> command allows you to reboot the FWSM and reload the configuration from a bootable floppy disk. If a disk is not present, it allows you to reboot and reload from the Flash partition.</p> <p>The FWSM does not accept abbreviations for <b>noconfirm</b>.</p> <p>You are prompted for confirmation before the “Proceed with reload?” message displays. Any response other than <b>n</b> causes the reboot to occur.</p>
------------------	---



### 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:
----------	--

```
fws(config)# reload
Proceed with reload? [confirm] y

Rebooting...

fws Bios V2.7
...
```

Related Commands	<b>shutdown</b>
------------------	-----------------

# rename

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

```
rename [disk:] [source-path] [disk:] [destination-path]
```

## Syntax Description

<b>disk:</b>	(Optional) Specifies the location of the source file.
<i>source-path</i>	(Optional) Specifies the path of the source file.
<b>disk:</b>	(Optional) Specifies the location of the destination file.
<i>destination-path</i>	(Optional) Specifies the path of the destination file.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system command line

Command Mode: privileged mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

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

## Examples

This example shows how to show the contents of a file named test1:

```
fwsd(config)# rename disk: disk:  
Source filename [running-config]? test  
Destination filename [n]? test1
```

**Related Commands**

cd  
copy disk  
copy flash  
copy startup-config  
copy tftp  
dir  
format  
mkdir  
more  
pwd  
rmdir  
show file

# rip

To enable and change Routing Information Protocol (RIP) settings, use the **rip** command. To disable the FWSM IP routing table updates, use the **no** form of this command.

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

```
no rip interface_name
```

## Syntax Description

<i>interface_name</i>	Internal or external network interface name.
<b>default</b>	Broadcasts a default route on the interface.
<b>passive</b>	Enables passive RIP on the interface.
<b>version</b>	(Optional) Specifies the RIP version; valid values are <b>1</b> and <b>2</b> .
<b>authentication</b>	(Optional) Enables RIP version 2 authentication.
<i>text</i>	(Optional) Sends RIP updates as clear text (not recommended).
<i>md5</i>	(Optional) Sends RIP updates using MD5 encryption.
<i>key</i>	(Optional) Key to encrypt RIP updates.
<i>key_id</i>	(Optional) Key identification value; valid values are from 1 to 255.

## Defaults

Enabled

## Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **rip** command allows you to enable IP routing table updates from received RIP broadcasts. If you specify RIP version 2, you can encrypt RIP updates using MD5 encryption. The **version 1** keyword provides backward compatibility with the older version.

Ensure that the *key* and *key\_id* arguments are the same that are used on any other device in your network that makes RIP version 2 updates. The *key* is a text string of up to 16 characters.

The FWSM cannot pass RIP updates between interfaces.

You configure RIP version 2 in passive mode. The FWSM listens for RIP routing broadcasts and uses that information to populate its routing tables. The FWSM accepts RIP version 2 multicast updates with an IP destination of 224.0.0.9. For RIP version 2 default mode, the FWSM transmits default route updates using an IP destination of 224.0.0.9. Configuring RIP version 2 registers the multicast address 224.0.0.9 so that the interface can accept multicast RIP version 2 updates.

Only Intel 10/100 and Gigabit interfaces support multicasting.

When you remove the RIP version 2 commands for an interface, you are unregistering the multicast address from the interface card.

### Examples

This example shows how to sample output from the version 1 **show rip** and **rip inside default** commands:

```
fwsM/context_name(config)# show rip
rip outside passive
no rip outside default
rip inside passive
no rip inside default

fwsM/context_name(config)# rip inside default
fwsM/context_name(config)# show rip
rip outside passive
no rip outside default
rip inside passive
rip inside default
```

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

- Enable version 2 passive 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.

```
fwsM/context_name(config)# rip outside passive version 2 authentication md5 thisisaKey 2
fwsM/context_name(config)# rip outside default version 2 authentication md5 thisisaKey 2
fwsM/context_name(config)# rip inside passive
fwsM/context_name(config)# rip dmz passive version 2

fwsM/context_name(config)# show 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
```

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

```
fwsM/context_name(config)# rip out default version 2 authentication text thisisaKey 3
fwsM/context_name(config)# show rip
rip outside default version 2 authentication text thisisaKey 3
```

### Related Commands

**clear rip**  
**show rip**

# rmdir

To remove the existing directory, use the **rmdir** command.

```
rmdir [disk:] [path]
```

## Syntax Description

<b>disk:</b>	(Optional) Changes the current working directory.
<i>path</i>	(Optional) Specifies the directory location.

## Defaults

This command has no default settings.

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: system command line  
 Command Mode: privileged mode  
 Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

If a file exists in the directory, the command fails. The **rmdir** command asks you for confirmation before removing the directory. The **rmdir disk:** command prompts you to enter the name of the directory that you are removing.

## Examples

This example shows how to remove an existing directory:

```
fwsm(config)# rmdir test
```

## Related Commands

- cd
- copy disk
- copy flash
- copy startup-config
- copy tftp
- dir
- format
- mkdir
- more
- pwd
- rename
- show file

# route

To enter a static or default route for the specified interface, use the **route** command. Use the **no** form of this command to remove routes from the specified interface.

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

## Syntax Description

<i>interface_name</i>	Internal or external network interface name.
<i>ip_address</i>	Internal or external network IP address.
<i>netmask</i>	Specifies a network mask to apply to <i>ip_address</i> .
<i>gateway_ip</i>	Specifies the IP address of the gateway router (the next-hop address for this route).
<i>metric</i>	(Optional) Specifies the number of hops to <i>gateway_ip</i> .

## Defaults

*metric* is 1.

## Command Modes

Security Context Mode: single context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode and transparent firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## 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.

If you are not sure about the number of hops to *gateway\_ip*, enter **1**. Your network administrator can supply this information or you can use a **traceroute** command to obtain the number of hops.

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.

```
fwsm/context_name(config)# route dmz 192.168.42.0 255.255.255.0 192.168.1.5 1
```

The routing table automatically specifies the IP address of a FWSM interface in the **route** command. Once you enter the IP address for each interface, the FWSM creates a **route** statement entry that is not deleted when you use the **clear route** command.

If the **route** command uses the IP address from one of the FWSM's interfaces 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**

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

```
fwsM/context_name(config)# route outside 0 0 209.165.201.1 1
```

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

```
fwsM/context_name(config)# route dmz1 10.1.2.0 255.0.0.0 10.1.1.4 1  
fwsM/context_name(config)# route dmz1 10.1.3.0 255.0.0.0 10.1.1.4 1
```

---

**Related Commands**

**clear route**  
**show route**

# route-map

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

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

## Syntax Description

<i>map_tag</i>	Text for the route map tag; the text can be up to 58 characters in length.
<b>permit</b>	(Optional) Specifies that if the match criteria is met for this route map, the route is redistributed as controlled by the set actions.
<b>deny</b>	(Optional) Specifies that if the match criteria are met for the route map, the route is not redistributed.
<i>seq_num</i>	(Optional) Route map sequence number; valid values are from 0 to 65535.

## 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

Security Context Mode: single context mode

Access Location: system and context command line

Command Mode: privileged mode

Transparent Mode: Routed

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The **route-map** command allows you to redistribute routes or to subject packets to policy routing.

The **route-map** global configuration command and the **match** and **set route-map** 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 give 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 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.

Another purpose of route maps is to enable policy routing. Use the **ip policy route-map** command, in addition to the **route-map** command, and the **match** and **set** commands to define the conditions for policy routing packets. The **match** commands specify the conditions under which policy routing occurs. The **set** commands specify the routing actions to perform if the criteria enforced by the **match** commands are met. You might want to specify policy route packets in a way other than the obvious shortest path.

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.

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

This example show how to configure a route map in OSPF routing:

```
fws#(config)# route-map maptag1 permit 8
fws#(config-route-map)# set metric 5
fws#(config-route-map)# match metric 5
fws#(config-route-map)# set metric-type type-2
fws#(config-route-map)# show route-map
route-map maptag1 permit 8
set metric 5
set metric-type type-2
match metric 5
fws#(config-route-map)# exit
fws#(config)#
```

**Related Commands**

**clear route-map**  
**match interface (route map submenu)**  
**match ip next-hop (route map submenu)**  
**match ip route-source (route map submenu)**  
**match metric (route map submenu)**  
**match route-type (route map submenu)**  
**set ip next-hop**  
**set ip next-hop (route map submenu)**  
**set metric**  
**set metric-type**  
**show route-map**

# router

To configure the router's IP address, use the **router** command. To remove the router ID, use the **no** form of this command.

[no] **router** *ip\_address*

Syntax Description	<i>ip_address</i>	Router ID in IP address format.
--------------------	-------------------	---------------------------------

**Defaults** This command has no default settings.

**Command Modes**

- Security Context Mode: single context mode
- Access Location: system and context command line
- Command Mode: configuration mode
- Transparent Mode: Routed

Command History	Release	Modification
	1.1(1)	Support for this command was introduced on the FWSM.

**Examples** This example shows how to configure the router's IP address:

```
fwsn(config)# router 122.34 45.10
```

**Related Commands** `show router`

# router-id

To configure the fixed router ID for an Open Shortest Path First (OSPF) process, use the **router-id** command. To use the previous OSPF router ID behavior, use the **no** form of this command to reset the OSPF.

**[no] router-id** *ip\_address*

## Syntax Description

*ip\_address* Router ID in IP address format.

This command has no default settings.

## Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode  
 Transparent Mode: Routed

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## 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 (DBDs). To prevent this situation, set the **router-id** *ip\_address* to a global address.

## Examples

This example shows how to configure the fixed router ID for OSPF:

```
fwsm(config)# router-id 123.45.46.10
```

## Related Commands

```
router ospf  

show ospf  

show routing  

show router-id
```

# router ospf

To enable OSPF routing through the FWSM, use the **router ospf** command. To terminate the OSPF routing process specified by its *pid*, use the **no** form of this command.

[no] **router ospf** *pid*

## Syntax Description

<i>pid</i>	Internally used identification parameter for an OSPF routing process; valid values are from 1 to 65534.
------------	---

## Defaults

OSPF routing is disabled on the FWSM.

## Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: context command line

Command Mode: configuration mode

Firewall Mode: routed firewall mode

## Command History

Release	Modification
1.1(1)	Support for this command was introduced on the FWSM.

## Usage Guidelines

The OSPF protocol is used instead of the Routing Information Protocol (RIP). Do not attempt to configure the FWSM for both OSPF and RIP at the same time.

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 the submode.

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*.

The **show ospf** command displays the configured **router ospf** subcommands.

You assign the *pid* locally on the firewall. You must assign a unique value for each OSPF routing process.

Once you enter the **route-ospf** command, the command prompt appears as (config-router)#, indicating that you are in the submode.

The **router ospf** command is used with the following OSPF-specific subcommands 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 type 7 default in the NSSA area.
- **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.
- **network**—Defines the interfaces on which OSPF runs and the area ID for those interfaces.
- **redistribute**—Configures the redistribution between OSPF processes according to the parameters specified.
- **router-id**—Creates a fixed router ID.
- **summary-address**—Creates the aggregate addresses for OSPF.
- **timers**—Configures the OSPF process delay timers.

---

**Examples**

This example shows how to enter the submode on the outside interface of the FWSM:

```
fwsm(config)# router ospf 5
```

---

**Related Commands**

**route-map**

**routing interface**

**show ip ospf**

See also the list of subcommands in the “Usage Guidelines” section.

# routing interface

To configure interface-specific Open Shortest Path First (OSPF) routing parameters, use the **routing interface** command. To remove the routing configuration for the interface specified only, use the **no** form of this command.

**[no] routing interface** *interface\_name*

---

## Syntax Description

*interface\_name* Name of the interface to configure.

---



---

## Defaults

OSPF routing is disabled on the FWSM interfaces.

---

## Command Modes

Security Context Mode: single context mode and multiple context mode  
 Access Location: context command line  
 Command Mode: configuration mode  
 Firewall Mode: routed firewall mode

---

## Usage Guidelines

The **routing interface** *interface\_name* command is the main command for all interface-specific OSPF interface mode commands. Enter this command with the name of the FWSM interface (*interface\_name*) that you want to configure, and then proceed with interface-specific configuration through the **routing interface** subcommands. You do not need to specify optional arguments in the **no** forms of the **routing interface** subcommands (unless they provide necessary information).

---

## Examples

This example shows how to enter the submode on the outside interface of the FWSM:

```
fwsm(config)# routing interface outside
```



### Note

In the routing submode, the command prompt appears as “(config-routing)#”.

---

This example shows the configuration for two concurrently running OSPF processes, with the IDs 5 and 12, on the outside interface of the FWSM:

```
fwsm(config)# routing interface  
fwsm(config)# show ospf
```

```
Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5  

Supports only single TOS(TOS0) routes  

Supports opaque LSA  

SPF schedule delay 5 secs, Hold time between two SPFs 10 secs  

Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs  

Number of external LSA 0. Checksum Sum 0x 0  

Number of opaque AS LSA 0. Checksum Sum 0x 0  

Number of DCbitless external and opaque AS LSA 0  

Number of DoNotAge external and opaque AS LSA 0  

Number of areas in this router is 0. 0 normal 0 stub 0 nssa  

External flood list length 0
```

```
Routing Process "ospf 12" with ID 172.23.59.232 and Domain ID 0.0.0.12
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x      0
Number of opaque AS LSA 0. Checksum Sum 0x      0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 0. 0 normal 0 stub 0 nssa
External flood list length 0
```

This example shows how to change the retransmit interval to 15 seconds:

```
fwsm(config)# ospf retransmit-interval 15
```

---

**Related Commands**

**ospf (interface submode)**  
**route-map**  
**router ospf**

## rpc-server

To create the remote processor call (RPC) services table, use the **rpc-server** command. To remove the RPC services table from the configuration, use the **no** form of this command.

```
[no] rpc-server ifc_name ip_addr mask service service_type protocol [TCP | UDP] port port
      [-port] timeout hh:mm:ss
```

```
no rpc-server active service service_type server ip_addr
```

### Syntax Description

<i>ifc_name</i>	Server interface name.
<i>ip_addr</i>	RPC server IP address.
<i>mask</i>	Network mask.
<b>service</b>	Specifies a service.
<i>service_type</i>	Sets the RPC service program number as specified in the <b>rpcinfo</b> command.
<b>protocol tcp</b>	Specifies the RPC transport protocol.
<b>protocol udp</b>	Specifies the RPC transport protocol.
<b>port port</b> [- port ]	Specifies the RPC protocol port range.
<b>port- port</b>	(Optional) Specifies the RPC protocol port range.
<b>timeout</b> <i>hh:mm:ss</i>	Specifies the timeout idle time after which the access for the RPC service traffic is closed.

### Defaults

This command has no default settings.

### Command Modes

Security Context Mode: single context mode  
 Access Location: system and context command line  
 Command Mode: configuration mode

### Command History

Release	Modification
2.2(1)	Support for this command was introduced on the FWSM.

### Examples

This example shows how to create an RPC services table:

```
fwsM/context_name(config)# rpc-server inside 30.26.0.23 255.255.0.0 service 2147483647
protocol TCP port 2222 timeout 0:03:00
```

### Related Commands

```
clear rpc-server
show rpc-server
```