



## CHAPTER 8

# Configuring IP Routing and DHCP Services

---

This chapter describes how to configure IP routing and DHCP on the FWSM. This chapter includes the following sections:

- [How Routing Behaves Within FWSM, page 8-1](#)
- [Configuring Static and Default Routes, page 8-2](#)
- [Defining a Route Map, page 8-5](#)
- [Configuring BGP Stub Routing, page 8-6](#)
- [Configuring OSPF, page 8-9](#)
- [Configuring RIP, page 8-21](#)
- [Configuring EIGRP, page 8-22](#)
- [Configuring Asymmetric Routing Support, page 8-30](#)
- [Configuring Route Health Injection, page 8-32](#)
- [Configuring DHCP, page 8-35](#)

## How Routing Behaves Within FWSM

FWSM uses both routing table and XLATE tables for routing decisions. To handle destination-ip-translated, that is, untranslated traffic, FWSM searches for existing XLATE, or static translation to select the egress interface. The selection process is as follows:

### Egress Interface Selection Process

- If destination-ip-translating XLATE already exists, the egress interface for the packet is determined from the XLATE table, but not from the routing table.
- If destination-ip-translating XLATE does not exist, but a matching static translation exists, then the egress interface is determined from the static route and an XLATE is created, and the routing table is not used.
- If destination-ip-translating XLATE does not exist and no matching static translation exists, the packet is not destination-ip-translated. FWSM processes this packet by looking up the route to select egress interface, then source-ip translation is performed (if necessary).

Therefore, for regular dynamic outbound NAT, initial outgoing packets are routed using the route table and then create the XLATE. Incoming return packets are forwarded using existing XLATEs only. For static NAT, destination-translated incoming packets are always forwarded using existing XLATE or static translation rules.

## Next Hop Selection Process

After selecting egress interface using any method described above, an additional route lookup is performed to find out suitable next hop(s) that belong to previously selected egress interface. If there are no routes in routing table that explicitly belong to selected interface, the packet is dropped with level 6 error message 110001 "no route to host", even if there is another route for a given destination network that belongs to different egress interface. If the route that belongs to selected egress interface is found, the packet is forwarded to corresponding next hop.

Load sharing on FWSM is possible only for multiple next-hops available using single egress interface. Load sharing cannot share multiple egress interfaces.

This is not true if the following conditions exist:

- If dynamic routing is in use on FWSM and route table changes after XLATE creation, for example a route flap happens, then destination-translated traffic is still forwarded using old XLATE, not via route table, until XLATE times out. It may be either forwarded to wrong interface or dropped with message 110001 "no route to host" if old route was removed from the old interface and attached to another one by routing process.
- The same problem may happen when there is no route flaps on FWSM itself, but some routing process is flapping around it, sending source-translated packets that belong to the same flow through FWSM using different interfaces. Destination-translated return packets may be forwarded back using the wrong egress interface.

This issue has a high probability in same-security-traffic configuration, where virtually any traffic may be either source-translated or destination-translated, depending on direction of initial packet in the flow.

When this issue occurs after a route flap, it can be resolved manually by using the **clear xlate** command, or automatically resolved by an XLATE timeout. XLATE timeout may be decreased if necessary. To ensure that this rarely happens, make sure that there is no route flaps on FWSM and around it. That is, ensure that destination-translated packets that belong to the same flow are always forwarded the same way through FWSM.

## Configuring Static and Default Routes

This section describes how to configure static and default routes on FWSM.

Multiple context mode does not support dynamic routing, so you must use static routes for any networks to which FWSM is not directly connected; for example, when there is a router between a network and FWSM.

You might want to use static routes in single context mode in the following cases:

- Your networks use a different router discovery protocol from RIP or OSPF.
- Your network is small and you can easily manage static routes.
- You do not want the traffic or CPU overhead associated with routing protocols.

The simplest option is to configure a default route to send all traffic to an upstream router, relying on the router to route the traffic for you. However, in some cases the default gateway might not be able to reach the destination network, so you must also configure more specific static routes. For example, if the default gateway is outside, then the default route cannot direct traffic to any inside networks that are not directly connected to FWSM.

In transparent firewall mode, for traffic that originates on FWSM and is destined for a non-directly connected network, you need to configure either a default route or static routes so FWSM knows out of which interface to send traffic. Traffic that originates on FWSM might include communications to a system log server, Websense or N2H2 server, or AAA server. If you have servers that cannot all be reached through a single default route, then you must configure static routes.

**Note**

The default route for the transparent firewall, which is required to provide a return path for management traffic, is only applied to management traffic from one bridge group network. This is because the default route specifies an interface in the bridge group as well as the router IP address on the bridge group network, and you can only define one default route. If you have management traffic from more than one bridge group network, you need to specify a static route that identifies the network from which you expect management traffic.

The FWSM supports up to three equal cost routes to the same destination per interface for load balancing.

This section includes the following topics:

- [Configuring a Static Route, page 8-3](#)
- [Configuring a Default Route, page 8-4](#)
- [Monitoring a Static or Default Route, page 8-5](#)

For information about configuring IPv6 static and default routes, see the “[Configuring IPv6 Default and Static Routes](#)” section on page 10-5.

## Configuring a Static Route

To add a static route, enter the following command:

```
hostname(config)# route if_name dest_ip mask gateway_ip [distance]
```

The *dest\_ip* and *mask* is the IP address for the destination network and the *gateway\_ip* is the address of the next-hop router.

The *distance* is the administrative distance for the route. The default is 1 if you do not specify a value. Administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connect routes. The default administrative distance for routes discovered by OSPF is 110. If a static route has the same administrative distance as a dynamic route, the static routes take precedence. Connected routes always take precedence over static or dynamically discovered routes.

Static routes remain in the routing table even if the specified gateway becomes unavailable. If the specified gateway becomes unavailable, you need to remove the static route from the routing table manually. However, static routes are removed from the routing table if the associated interface goes down. They are reinstated when the interface comes back up.

**Note**

If you create a static route with an administrative distance greater than the administrative distance of the routing protocol running on the FWSM, then a route to the specified destination discovered by the routing protocol takes precedence over the static route. The static route is used only if the dynamically discovered route is removed from the routing table.

The following example creates a static route that sends all traffic destined for 10.1.1.0/24 to the router (10.1.2.45) connected to the inside interface:

```
hostname(config)# route inside 10.1.1.0 255.255.255.0 10.1.2.45 1
```

You can define up to three equal cost routes to the same destination per interface. ECMP is not supported across multiple interfaces. With ECMP, the traffic is not necessarily divided evenly between the routes; traffic is distributed among the specified gateways based on an algorithm that hashes the source and destination IP addresses.

The following example shows static routes that are equal cost routes that direct traffic to three different gateways on the outside interface. The FWSM distributes the traffic among the specified gateways.

```
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.2
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.3
```

## Configuring a Default Route

A default route identifies the gateway IP address to which FWSM sends all IP packets for which it does not have a learned or static route. A default route is simply a static route with 0.0.0.0/0 as the destination IP address. Routes that identify a specific destination take precedence over the default route.

You can define up to three equal cost default route entries per device. Defining more than one equal cost default route entry causes the traffic sent to the default route to be distributed among the specified gateways. When defining more than one default route, you must specify the same interface for each entry.

If you attempt to define more than three equal cost default routes, or if you attempt to define a default route with a different interface than a previously defined default route, you receive the message “ERROR: Cannot add route entry, possible conflict with existing routes.”

To define the default route, enter the following command:

```
hostname(config)# route if_name 0.0.0.0 0.0.0.0 gateway_ip [distance]
```

**Tip**

You can enter 0 0 instead of 0.0.0.0 0.0.0.0 for the destination network address and mask, for example:

```
hostname(config)# route outside 0 0 192.168.1 1
```

The following example shows an FWSM configured with three equal cost default routes. Traffic received by the FWSM for which there is no static or learned route is distributed among the gateways with the IP addresses 192.168.2.1, 192.168.2.2, 192.168.2.3.

```
hostname(config)# route outside 0 0 192.168.2.1
hostname(config)# route outside 0 0 192.168.2.2
hostname(config)# route outside 0 0 192.168.2.3
```

## Monitoring a Static or Default Route



**Note**

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

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

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

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

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

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

To monitor a static or default route, and to switch to an alternate path in the event a router goes down, use the Command Line Interface tool to enter the following command.

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

## Defining a Route Map

Route maps are used to redistribute routes between processes or for route health injection (RHI). To define a route map for use with supported features, perform the following steps:

**Step 1** To create a route map entry, enter the following command:

```
hostname(config)# route-map name {permit | deny} [sequence_number]
```

Route map entries are read in order. You can identify the order using the *sequence\_number* option, or the FWSM uses the order in which you add the entries.

**Step 2** Enter one or more **match** commands:

- To match any routes that have a destination network that matches a standard access list, enter the following command:

```
hostname(config-route-map)# match ip address acl_id [acl_id] [...]
```

If you specify more than one access list, then the route can match any of the access lists.

- To match any routes that have a specified metric, enter the following command:

```
hostname(config-route-map)# match metric metric_value
```

The *metric\_value* can be from 0 to 4294967295.

- To match any routes that have a next hop router address that matches a standard access list, enter the following command:

```
hostname(config-route-map)# match ip next-hop acl_id [acl_id] [...]
```

If you specify more than one access list, then the route can match any of the access lists.

- To match any routes with the specified next hop interface, enter the following command:

```
hostname(config-route-map)# match interface if_name
```

If you specify more than one interface, then the route can match either interface.

- To match any routes that have been advertised by routers that match a standard access list, enter the following command:

```
hostname(config-route-map)# match ip route-source acl_id [acl_id] [...]
```

If you specify more than one access list, then the route can match any of the access lists.

- To match the route type, enter the following command:

```
hostname(config-route-map)# match route-type {internal | external [type-1 | type-2]}
```

### Step 3 Enter one or more **set** commands.

If a route matches the **match** commands, then the following **set** commands determine the action to perform on the route before redistributing it.

- To set the metric, enter the following command:

```
hostname(config-route-map)# set metric metric_value
```

The *metric\_value* can be a value between 0 and 294967295

- To set the metric type, enter the following command:

```
hostname(config-route-map)# set metric-type {type-1 | type-2}
```

---

The following example shows how to redistribute routes with a hop count equal to 1. The FWSM redistributes these routes as external LSAs with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
```

## Configuring BGP Stub Routing

The FWSM supports BGP stub routing. The BGP stub routing process advertises static and directly connected routes but does not accept routes advertised by the BGP peer.

BGP stub routing is a licensed feature. You must have or obtain a license key that supports BGP stub routing to configure this feature.

This section includes the following topics:

- [BGP Stub Limitations, page 8-7](#)
- [Configuring BGP Stub Routing, page 8-7](#)
- [Monitoring BGP Stub Routing, page 8-8](#)
- [Restarting the BGP Stub Routing Process, page 8-9](#)

## BGP Stub Limitations

The following limitations apply to configuring BGP stub routing on the FWSM:

- You can only configure one BGP routing process, even in multiple context mode.
- You can only configure one BGP neighbor, even in multiple context mode.
- The FWSM does not process UPDATE messages received from the BGP neighbor. It can only send routing updates to the BGP neighbor.
- The FWSM only advertises static routes and directly-connected networks. You cannot redistribute routes from other routing protocols into the BGP routing process.
- In multiple context mode, the FWSM can only advertise static routes and directly-connected networks for the context that contains the interface the BGP peer is reachable through and for which there are configured **network** commands. If the BGP neighbor is reachable through an interface that is shared across multiple contexts, then all of the static routes and directly-connected networks in the contexts sharing the interface are available to the BGP routing process.
- BGP stub does not support IPv6, VPN, or NLRI multicast.
- Only iBGP is supported; eBGP is not supported.

## Configuring BGP Stub Routing

Before configuring BGP stub routing on the FWSM:

- You must enable route reflector on the BGP neighbor.
- If the FWSM is in multiple context mode, you must be in the admin context to configure BGP stub routing. Additionally, the admin context must be in routed mode.



### Note

---

Although in multiple context mode the BGP routing process is configured in the admin context, only the static routes and directly-connected networks for the context that the BGP peer is reachable through can be advertised.

---

To enable and configure a BGP routing process, perform the following steps:

---

**Step 1** Create the BGP routing process by entering the following command:

```
hostname(config)# router bgp as-number
```

The *as-number* argument is the autonomous system number that identifies the FWSM to other BGP routers and tags the routing information passed along. It must be the same as the AS number of the BGP neighbor. After entering this command, the command prompt changes to `hostname(config-router)#` to indicate that you are now in router configuration mode for the specified routing process.

- Step 2** (Optional) Specify the router ID for the FWSM by entering the following command. If you do not enter a router ID, the highest IP address configured on the FWSM is used.

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

The *id* can be any IP address, including an IP address that is not configured on the FWSM. If this command is not specified, the router ID used is the highest IP address configured on the FWSM.

- Step 3** Specify the BGP neighbor that BGP updates are sent to by entering the following command:

```
hostname(config-router)# neighbor ip-addr remote-as as-number
```

The *ip-addr* argument is the IP address of the BGP neighbor. The *as-number* is the autonomous system number of the BGP neighbor. This should be the same as the AS number configured on the FWSM with the **router** command.

- Step 4** (Optional) Enter the password used to authenticate the BGP message to the neighbor. This password must be set on both the neighbor and the FWSM before BGP messages can be exchanged.

```
hostname(config-router)# neighbor ip-addr password [mode] password
```

The *ip-addr* argument is the IP address of the BGP neighbor defined with the **neighbor** command. The *mode* argument can be from 0 to 7. If used, the BGP neighbor must use the same mode. The *password* argument is an alphanumeric string that can contain keyboard symbols but cannot contain spaces.

- Step 5** Specify the networks that the BGP routing process advertises using the **network** command. You can configure up to 200 network commands on the FWSM.

```
hostname(config-router)# network ip-addr mask mask
```

The BGP stub routing process only advertises static and directly-connected networks. The **network** command defines which of those networks are advertised in BGP updates.

## Monitoring BGP Stub Routing

You can use the following commands to display information about the BGP routing process, neighbor, and advertised routes. In multiple context mode, these commands are entered in the admin context.

- To display information about the BGP routing process, enter the following command:

```
hostname# show ip bgp summary
```

- To display BGP neighbor information, enter the following command:

```
hostname# show ip bgp neighbors
```

- To display the routes advertised by the BGP routing process, enter the following command:

```
hostname# show ip bgp neighbors advertised-routes
```

- To view debug messages for the BGP routing process, enter the following command:

```
hostname# debug ip bgp
```

For more detailed information about the output from these commands, see the command information in the *Catalyst 6500 Series and Cisco 7600 Series Switch Firewall Services Module Command Reference*.

## Restarting the BGP Stub Routing Process

To clear the BGP session established with the neighbor, clear the statistical counters associated with the session, and restart the BGP with the neighbor, enter the following command:

```
hostname(config)# clear ip bgp neighbor-addr
```

## Configuring OSPF

This section describes how to configure OSPF. This section includes the following topics:

- [OSPF Overview, page 8-9](#)
- [Enabling OSPF, page 8-10](#)
- [Redistributing Routes Between OSPF Processes, page 8-11](#)
- [Configuring OSPF Interface Parameters, page 8-12](#)
- [Configuring OSPF Area Parameters, page 8-14](#)
- [Configuring OSPF NSSA, page 8-15](#)
- [Configuring a Point-To-Point, Non-Broadcast OSPF Neighbor, page 8-16](#)
- [Configuring Route Summarization Between OSPF Areas, page 8-17](#)
- [Configuring Route Summarization when Redistributing Routes into OSPF, page 8-17](#)
- [Generating a Default Route, page 8-18](#)
- [Configuring Route Calculation Timers, page 8-18](#)
- [Logging Neighbors Going Up or Down, page 8-19](#)
- [Displaying OSPF Update Packet Pacing, page 8-19](#)
- [Monitoring OSPF, page 8-20](#)
- [Restarting the OSPF Process, page 8-21](#)

## OSPF Overview

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly rather than gradually as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. FWSM calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

FWSM can run two processes of OSPF protocol simultaneously, on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside, and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

Redistribution between the two OSPF processes is supported. Static and connected routes configured on OSPF-enabled interfaces on FWSM can also be redistributed into the OSPF process. You cannot enable RIP on FWSM if OSPF is enabled. Redistribution between RIP and OSPF is not supported.

FWSM supports the following OSPF features:

- Support of intra-area, interarea, and external (Type I and Type II) routes.
- Support of a virtual link.
- OSPF LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring FWSM as a designated router or a designated backup router. FWSM also can be set up as an ABR; however, the ability to configure the FWSM as an ASBR is limited to default information only (for example, injecting a default route).
- Support for stub areas and not-so-stubby-areas.
- Area boundary router type-3 LSA filtering.
- Advertisement of static and global address translations.

## Enabling OSPF

To enable OSPF, you need to create an OSPF routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.



**Note** You cannot enable OSPF if RIP is enabled.

To enable OSPF, perform the following steps:

**Step 1** To create an OSPF routing process, enter the following command:

```
hostname(config)# router ospf process_id
```

This command enters the router configuration mode for this OSPF process.

The *process\_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

**Step 2** To define the IP addresses on which OSPF runs and to define the area ID for that interface, enter the following command:

```
hostname(config-router)# network ip_address mask area area_id
```

The following example shows how to enable OSPF:

```
hostname(config)# router ospf 2
```

```
hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0
```

## Redistributing Routes Between OSPF Processes

The FWSM can control the redistribution of routes between OSPF routing processes. The FWSM matches and changes routes according to settings in the **redistribute** command or by using a route map. See also the “[Generating a Default Route](#)” section on page 8-18 for another use for route maps.



### Note

Note: The maximum number of route entries for all types of routes (connected, static and dynamic) supported by FWSM is 32768, or 32K.

To redistribute static, connected, or OSPF routes from one process into another OSPF process, perform the following steps:

- Step 1** To create a route map, see the “[Defining a Route Map](#)” section on page 8-5.
- Step 2** If you have not already done so, enter the router configuration mode for the OSPF process you want to redistribute into by entering the following command:

```
hostname(config)# router ospf process_id
```

- Step 3** To specify the routes you want to redistribute, enter the following command:

```
hostname(config-router)# redistribute {ospf process_id
[match {internal | external 1 | external 2}] | static | connect} [metric metric-value]
[metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
```

The **ospf process\_id**, **static**, and **connect** keywords specify from where you want to redistribute routes.

You can either use the options in this command to match and set route properties, or you can use a route map. The **tag** and **subnets** options do not have equivalents in the **route-map** command. If you use both a route map and options in the **redistribute** command, then they must match.

The following example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The FWSM redistributes these routes as external LSAs with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
hostname(config-route-map)# set tag 1
hostname(config-route-map)# router ospf 2
hostname(config-router)# redistribute ospf 1 route-map 1-to-2
```

The following example shows the specified OSPF process routes being redistributed into OSPF process 109. The OSPF metric is remapped to 100.

```
hostname(config)# router ospf 109
hostname(config-router)# redistribute ospf 108 metric 100 subnets
```

The following example shows route redistribution where the link-state cost is specified as 5 and the metric type is set to external, indicating that it has lower priority than internal metrics.

```
hostname(config)# router ospf 1
```

```
hostname(config-router)# redistribute ospf 2 metric 5 metric-type external
```

## Configuring OSPF Interface Parameters

You can alter some interface-specific OSPF parameters as necessary. You are not required to alter any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key**. Be sure that if you configure any of these parameters, the configurations for all routers on your network have compatible values.

To configure OSPF interface parameters, perform the following steps:

---

**Step 1** To enter the interface configuration mode, enter the following command:

```
hostname(config)# interface if_name
```

**Step 2** Enter any of the following commands:

- To specify the authentication type for an interface, enter the following command:

```
hostname(config-interface)# ospf authentication [message-digest | null]
```

- To assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication, enter the following command:

```
hostname(config-interface)# ospf authentication-key key
```

The *key* can be any continuous string of characters up to 8 bytes in length.

The password created by this command is used as a key that is inserted directly into the OSPF header when the FWSM software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

- To explicitly specify the cost of sending a packet on an OSPF interface, enter the following command:

```
hostname(config-interface)# ospf cost cost
```

The *cost* is an integer from 1 to 65535.

- To set the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet, enter the following command:

```
hostname(config-interface)# ospf dead-interval seconds
```

The value must be the same for all nodes on the network.

- To specify the length of time between the hello packets that the FWSM sends on an OSPF interface, enter the following command:

```
hostname(config-interface)# ospf hello-interval seconds
```

The value must be the same for all nodes on the network.

- To enable OSPF MD5 authentication, enter the following command:

```
hostname(config-interface)# ospf message-digest-key key_id md5 key
```

Set the following values:

- *key\_id*—An identifier in the range from 1 to 255.

- *key*—Alphanumeric password of up to 16 bytes.

Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value.

We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.

- To set the priority to help determine the OSPF designated router for a network, enter the following command:

```
hostname(config-interface)# ospf priority number_value
```

The *number\_value* is between 0 to 255.

- To specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface, enter the following command:

```
hostname(config-interface)# ospf retransmit-interval seconds
```

The *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 65535 seconds. The default is 5 seconds.

- To set the estimated number of seconds required to send a link-state update packet on an OSPF interface, enter the following command:

```
hostname(config-interface)# ospf transmit-delay seconds
```

The *seconds* is from 1 to 65535 seconds. The default is 1 second.

The following example shows how to configure the OSPF interfaces:

```
hostname(config)# router ospf 2
hostname(config-router)# network 10.1.1.0 255.255.255.0 area 0
hostname(config-router)# interface inside
hostname(config-interface)# ospf cost 20
hostname(config-interface)# ospf retransmit-interval 15
hostname(config-interface)# ospf transmit-delay 10
hostname(config-interface)# ospf priority 20
hostname(config-interface)# ospf hello-interval 10
hostname(config-interface)# ospf dead-interval 40
hostname(config-interface)# ospf authentication-key cisco
hostname(config-interface)# ospf message-digest-key 1 md5 cisco
hostname(config-interface)# ospf authentication message-digest
```

The following is sample output from the **show ospf** command:

```
hostname(config)# show ospf

Routing Process "ospf 2" with ID 20.1.1.89.2 and Domain ID 0.0.0.2
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 5. Checksum Sum 0x 26da6
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 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
```

```

Number of interfaces in this area is 1
Area has no authentication
SPF algorithm executed 2 times
Area ranges are
Number of LSA 5. Checksum Sum 0x 209a3
Number of opaque link LSA 0. Checksum Sum 0x      0
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

```

## Configuring OSPF Area Parameters

You can configure several area parameters. These area parameters (shown in the following task table) include setting authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR, into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in the stub area. To further reduce the number of LSAs sent into a stub area, you can configure the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending summary link advertisement (LSA type 3) into the stub area.

To specify area parameters for your network, perform the following steps:

---

**Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

**Step 2** Enter any of the following commands:

- To enable authentication for an OSPF area, enter the following command:

```
hostname(config-router)# area area-id authentication
```

- To enable MD5 authentication for an OSPF area, enter the following command:

```
hostname(config-router)# area area-id authentication message-digest
```

- To define an area to be a stub area, enter the following command:

```
hostname(config-router)# area area-id stub [no-summary]
```

- To assign a specific cost to the default summary route used for the stub area, enter the following command:

```
hostname(config-router)# area area-id default-cost cost
```

The *cost* is an integer from 1 to 65535. The default is 1.

---

The following example shows how to configure the OSPF area parameters:

```

hostname(config)# router ospf 2
hostname(config-router)# area 0 authentication
hostname(config-router)# area 0 authentication message-digest
hostname(config-router)# area 17 stub

```

```
hostname(config-router)# area 17 default-cost 20
```

## Configuring OSPF NSSA

The OSPF implementation of an NSSA is similar to an OSPF stub area. NSSA does not flood type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

NSSA imports type 7 autonomous system external routes within an NSSA area by redistribution. These type 7 LSAs are translated into type 5 LSAs by NSSA ABRs, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

You can simplify administration if you are an ISP or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol using NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

To specify area parameters for your network as needed to configure OSPF NSSA, perform the following steps:

---

**Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

**Step 2** Enter any of the following commands:

- To define an NSSA area, enter the following command:

```
hostname(config-router)# area area-id nssa [no-redistribution]  
[default-information-originate]
```

- To summarize groups of addresses, enter the following command:

```
hostname(config-router)# summary address ip_address mask [not-advertise] [tag tag]
```

This command helps reduce the size of the routing table. Using this command for OSPF causes an OSPF ASBR to advertise one external route as an aggregate for all redistributed routes that are covered by the address.

OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

In the following example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement.

```
hostname(config-router)# summary-address 10.1.1.0 255.255.0.0
```

Before you use this feature, consider these guidelines:

- You can set a type 7 default route that can be used to reach external destinations. When configured, the router generates a type 7 default into the NSSA or the NSSA area boundary router.

- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate.

## Configuring a Point-To-Point, Non-Broadcast OSPF Neighbor

You need to define a static OSPF neighbor to advertise OSPF routes over a point-to-point, non-broadcast network. When an interface is configured as point-to-point, the following restrictions apply:

- You can define only one OSPF neighbor for the interface.
- You need to define a static route pointing to the OSPF neighbor if it is not on a directly connected network.
- The interface cannot form adjacencies unless neighbors are configured explicitly.

To define an OSPF neighbor on a point-to-point, non-broadcast network, perform the following tasks:

**Step 1** If the OSPF neighbor is not on a directly-connected network, create a static route to the OSPF neighbor. Do not use the default route. See the “[Configuring a Static Route](#)” section on page 8-3 for more information about creating static routes.

**Step 2** Define the OSPF neighbor by performing the following tasks:

- a.** Enter router configuration mode for the OSPF process. Enter the following command:

```
hostname(config)# router ospf pid
```

- b.** Define the OSPF neighbor by entering the following command:

```
hostname(config-router)# neighbor addr [interface if_name]
```

The *addr* argument is the IP address of the OSPF neighbor. The *if\_name* is the interface used to communicate with the neighbor. If the OSPF neighbor is not on the same network as any of the directly-connected interfaces, you must specify the **interface**.

- c.** If not already configured, define the networks and associated area ID for the interface facing the OSPF neighbor by entering the following command:

```
hostname(config-router)# network addr mask area area_id
```

The *addr mask* pair must cover the IP address of the interface.

**Step 3** Configure the interface through which the FWSM communicates with the neighbor by entering the following commands:

```
hostname(config)# interface vlan
hostname(config-if)# ospf network point-to-point non-broadcast
```

The following example shows how to configure OSPF across a point-to-point, non-broadcast network. The OSPF neighbor is not on a directly-connected network, so a static route is needed.

```
hostname(config)# route ospf_outside 10.3.3.0 255.255.255.0 10.1.1.99 1

hostname(config)# interface Vlan55
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
hostname(config-if)# ospf network point-to-point non-broadcast
```

```
hostname(config-if)# exit

hostname(config)# router ospf 1
hostname(config-router)# network 10.1.1.0 255.255.255.0 area 100
hostname(config-router)# neighbor 10.3.3.1 interface outside
hostname(config-router)# log-adj-changes
```

## Configuring Route Summarization Between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router advertises networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the area boundary router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To define an address range for route summarization, perform the following steps:

- 
- Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

- Step 2** To set the address range, enter the following command:

```
hostname(config-router)# area area-id range ip-address mask [advertise | not-advertise]
```

---

The following example shows how to configure route summarization between OSPF areas:

```
hostname(config)# router ospf 1
hostname(config-router)# area 17 range 12.1.0.0 255.255.0.0
```

## Configuring Route Summarization when Redistributing Routes into OSPF

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the FWSM to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

To configure the software advertisement on one summary route for all redistributed routes covered by a network address and mask, perform the following steps:

- 
- Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

- Step 2** To set the summary address, enter the following command:

```
hostname(config-router)# summary-address ip_address mask [not-advertise] [tag tag]
```

OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

---

The following example shows how to configure route summarization. The summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

```
hostname(config)# router ospf 1
hostname(config-router)# summary-address 10.1.0.0 255.255.0.0
```

## Generating a Default Route

You can force an ASBR to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the router automatically becomes an ASBR. However, an ASBR does not by default generate a default route into the OSPF routing domain.

To generate a default route, perform the following steps:

- 
- Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

- Step 2** To force the ASBR to generate a default route, enter the following command:

```
hostname(config-router)# default-information originate [always] [metric metric-value]
[metric-type {1 | 2}] [route-map map-name]
```

---

The following example shows how to generate a default route:

```
hostname(config)# router ospf 2
hostname(config-router)# default-information originate always
```

## Configuring Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts an SPF calculation. You also can configure the hold time between two consecutive SPF calculations.

To configure route calculation timers, perform the following steps:

- 
- Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

- Step 2** To configure the route calculation time, enter the following command:

```
hostname(config-router)# timers spf spf-delay spf-holdtime
```

The *spf-delay* is the delay time (in seconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 65535. The default time is 5 seconds. A value of 0 means that there is no delay; that is, the SPF calculation is started immediately.

The *spf-holdtime* is the minimum time (in seconds) between two consecutive SPF calculations. It can be an integer from 0 to 65535. The default time is 10 seconds. A value of 0 means that there is no delay; that is, two SPF calculations can be done, one immediately after the other.

The following example shows how to configure route calculation timers:

```
hostname(config)# router ospf 1
hostname(config-router)# timers spf 10 120
```

## Logging Neighbors Going Up or Down

By default, the system sends a system log message when an OSPF neighbor goes up or down.

Configure this command if you want to know about OSPF neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** router configuration command provides a higher level view of the peer relationship with less output. Configure **log-adj-changes detail** if you want to see messages for each state change.

To log neighbors going up or down, perform the following steps:

**Step 1** If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

**Step 2** To configure logging for neighbors going up or down, enter the following command:

```
hostname(config-router)# log-adj-changes [detail]
```



**Note** Logging must be enabled for the neighbor up/down messages to be sent.

The following example shows how to log neighbors up/down messages:

```
hostname(config)# router ospf 1
hostname(config-router)# log-adj-changes detail
```

## Displaying OSPF Update Packet Pacing

OSPF update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing packets might be dropped if either of the following topologies exist:

- A fast router is connected to a slower router over a point-to-point link.
- During flooding, several neighbors send updates to a single router at the same time.

Pacing is also used between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out an interface. The benefit of the pacing is that OSPF update and retransmission packets are sent more efficiently.

There are no configuration tasks for this feature; it occurs automatically.

To observe OSPF packet pacing by displaying a list of LSAs waiting to be flooded over a specified interface, enter the following command:

```
hostname# show ospf flood-list if_name
```

## Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can use the information provided to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To display various routing statistics, perform one of the following tasks, as needed:

- To display general information about OSPF routing processes, enter the following command:

```
hostname# show ospf [process-id [area-id]]
```

- To display the internal OSPF routing table entries to the ABR and ASBR, enter the following command:

```
hostname# show ospf border-routers
```

- To display lists of information related to the OSPF database for a specific router, enter the following command:

```
hostname# show ospf [process-id [area-id]] database
```

- To display a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing), enter the following command:

```
hostname# show ospf flood-list if-name
```

- To display OSPF-related interface information, enter the following command:

```
hostname# show ospf interface [if_name]
```

- To display OSPF neighbor information on a per-interface basis, enter the following command:

```
hostname# show ospf neighbor [if-name] [neighbor-id] [detail]
```

- To display a list of all LSAs requested by a router, enter the following command:

```
hostname# show ospf request-list neighbor if_name
```

- To display a list of all LSAs waiting to be resent, enter the following command:

```
hostname# show ospf retransmission-list neighbor if_name
```

- To display a list of all summary address redistribution information configured under an OSPF process, enter the following command:

```
hostname# show ospf [process-id] summary-address
```

- To display OSPF-related virtual links information, enter the following command:

```
hostname# show ospf [process-id] virtual-links
```

## Restarting the OSPF Process

To restart an OSPF process, clear redistribution, or counters, enter the following command:

```
hostname(config)# clear ospf pid {process | redistribution | counters  
[neighbor [neighbor-interface] [neighbor-id]]}
```

## Configuring RIP

This section describes how to configure RIP. This section includes the following topics:

- [RIP Overview, page 8-21](#)
- [Enabling RIP, page 8-21](#)

### RIP Overview

Devices that support RIP send routing-update messages at regular intervals and when the network topology changes. These RIP packets contain information about the networks that the devices can reach, as well as the number of routers or gateways that a packet must travel through to reach the destination address. RIP generates more traffic than OSPF, but is easier to configure initially.

RIP has advantages over static routes because the initial configuration is simple, and you do not need to update the configuration when the topology changes. The disadvantage to RIP is that there is more network and processing overhead than static routing. Additionally, RIP cannot be enabled on a global basis.

FWSM uses a limited version of RIP; it does not send out RIP updates that identify the networks that FWSM can reach. However, you can enable one or both of the following methods:

- **Passive RIP**—FWSM listens for RIP updates but does not send any updates about its networks out of the interface.

Passive RIP allows FWSM to learn about networks to which it is not directly connected.

- **Default Route Updates**—Instead of sending normal RIP updates that describe all the networks reachable through FWSM, FWSM sends a default route to participating devices that identifies FWSM as the default gateway.

You can use the default route option with passive RIP, or alone. You might use the default route option alone if you use static routes on FWSM, but do not want to configure static routes on downstream routers. Typically, you would not enable the default route option on the outside interface, because FWSM is not typically the default gateway for the upstream router.

### Enabling RIP

To enable RIP on an interface, enter the following command:

```
hostname(config)# rip if_name {default | passive} [version {1 | 2  
[authentication {text | md5} key key_id]]}
```

You can enable both the passive and default modes of RIP on an interface by entering the **rip** command twice, one time for each method. For example, enter the following commands:

```
hostname(config)# rip inside default version 2 authentication md5 scorpious 1
```

```
hostname(config)# rip inside passive version 2 authentication md5 scorpius 1
```

If you want to enable passive RIP on all interfaces, but only enable default routes on the inside interface, enter the following commands:

```
hostname(config)# rip inside default version 2 authentication md5 scorpius 1
hostname(config)# rip inside passive version 2 authentication md5 scorpius 1
hostname(config)# rip outside passive version 2 authentication md5 scorpius 1
```

**Note**

Before testing your configuration, flush the ARP caches on any routers connected to the FWSM. For Cisco routers, use the **clear arp** command to flush the ARP cache.

You cannot enable RIP if OSPF is enabled.

## Configuring EIGRP

This section describes the configuration and monitoring of EIGRP routing and includes the following topics:

- [EIGRP Routing Overview, page 8-22](#)
- [Enabling and Configuring EIGRP Routing, page 8-23](#)
- [Enabling and Configuring EIGRP Stub Routing, page 8-24](#)
- [Enabling EIGRP Authentication, page 8-25](#)
- [Defining an EIGRP Neighbor, page 8-26](#)
- [Redistributing Routes Into EIGRP, page 8-26](#)
- [Configuring the EIGRP Hello Interval and Hold Time, page 8-27](#)
- [Disabling Automatic Route Summarization, page 8-27](#)
- [Configuring Summary Aggregate Addresses, page 8-28](#)
- [Disabling EIGRP Split Horizon, page 8-28](#)
- [Changing the Interface Delay Value, page 8-29](#)
- [Monitoring EIGRP, page 8-29](#)
- [Disabling Neighbor Change and Warning Message Logging, page 8-30](#)

## EIGRP Routing Overview

EIGRP is an enhanced version of IGRP developed by Cisco. Unlike IGRP and RIP, EIGRP does not send out periodic route updates. EIGRP updates are sent out only when the network topology changes.

Neighbor discovery is the process that the FWSM uses to dynamically learn of other routers on directly attached networks. EIGRP routers send out multicast hello packets to announce their presence on the network. When the FWSM receives a hello packet from a new neighbor, it sends its topology table to the neighbor with an initialization bit set. When the neighbor receives the topology update with the initialization bit set, the neighbor sends its topology table back to the FWSM.

The hello packets are sent out as multicast messages. No response is expected to a hello message. The exception to this is for statically defined neighbors. If you use the **neighbor** command to configure a neighbor, the hello messages sent to that neighbor are sent as unicast messages. Routing updates and acknowledgements are sent out as unicast messages.

Once this neighbor relationship is established, routing updates are not exchanged unless there is a change in the network topology. The neighbor relationship is maintained through the hello packets. Each hello packet received from a neighbor contains a hold time. This is the time in which the FWSM can expect to receive a hello packet from that neighbor. If the FWSM does not receive a hello packet from that neighbor within the hold time advertised by that neighbor, the FWSM considers that neighbor to be unavailable.

The EIGRP uses an algorithm called DUAL for route computations. DUAL saves all routes to a destination in the topology table, not just the least-cost route. The least-cost route is inserted into the routing table. The other routes remain in the topology table. If the main route fails, another route is chosen from the feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination. The feasibility calculation guarantees that the path is not part of a routing loop.

If a feasible successor is not found in the topology table, a route recomputation must occur. During route recomputation, DUAL queries the EIGRP neighbors for a route, who in turn query their neighbors. Routers that do not have a feasible successor for the route return an unreachable message.

During route recomputation, DUAL marks the route as active. By default, the FWSM waits for three minutes to receive a response from its neighbors. If the FWSM does not receive a response from a neighbor, the route is marked as stuck-in-active. All routes in the topology table that point to the unresponsive neighbor as a feasibility successor are removed.

## Enabling and Configuring EIGRP Routing

You can only enable one EIGRP routing process on the FWSM.

To enable and configure EIGRP routing, perform the following tasks:

- Step 1** Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

- Step 2** To configure the interfaces and networks that participate in EIGRP routing, configure one or more **network** statements by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

Directly-connected networks that fall within the defined network are advertised by the FWSM. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to participate in EIGRP routing, but that is attached to a network that you want advertised, configure a **network** command that covers the network the interface is attached to, and use the **passive-interface** command to prevent that interface from sending or receiving EIGRP updates.

- Step 3** (Optional) To prevent an interface from sending or receiving EIGRP routing message, enter the following command:

```
hostname(config-router)# passive-interface {default | if-name}
```

Using the **default** keyword disables EIGRP routing updates on all interfaces. Specifying an interface name, as defined by the **nameif** command, disables EIGRP routing updates on the specified interface. You can have multiple **passive-interface** commands in your EIGRP router configuration.

- Step 4** (Optional) To control the sending or receiving of candidate default route information, enter the following command:

```
hostname(config-router)# no default-information {in | out}
```

Configuring **no default-information in** causes the candidate default route bit to be blocked on received routes. Configuring **no default-information out** disables the setting of the default route bit in advertised routes.

- Step 5** (Optional) To filter networks sent in EIGRP routing updates, perform the following steps:

- a. Create a standard access list that defines the routes you want to advertise.
- b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates sent by that interface.

```
hostname(config-router): distribute-list acl out [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

- Step 6** (Optional) To filter networks received in EIGRP routing updates, perform the following steps:

- a. Create a standard access list that defines the routes you want to filter from received updates.
- b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates received by that interface.

```
hostname(config-router): distribute-list acl in [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

## Enabling and Configuring EIGRP Stub Routing

You can configure the FWSM as an EIGRP stub router. Stub routing decreases memory and processing requirements on the FWSM. As a stub router, the FWSM does not need to maintain a complete EIGRP routing table because it forwards all nonlocal traffic to a distribution router. Generally, the distribution router need not send anything more than a default route to the stub router.

Only specified routes are propagated from the stub router to the distribution router. As a stub router, the FWSM responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” When the FWSM is configured as a stub, it sends a special peer information packet to all neighboring routers to report its status as a stub router. Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

To enable and configure and EIGRP stub routing process, perform the following steps:

- Step 1** Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

- Step 2** Configure the interface connected to the distribution router to participate in EIGRP by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

- Step 3** Configure the stub routing process by entering the following command. You must specify which networks are advertised by the stub routing process to the distribution router. Static and connected networks are not automatically redistributed into the stub routing process.

```
hostname(config-router)# eigrp stub {receive-only | [connected] [redistributed] [static] [summary]}
```

---

## Enabling EIGRP Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

EIGRP route authentication is configured on a per-interface basis. All EIGRP neighbors on interfaces configured for EIGRP message authentication must be configured with the same authentication mode and key for adjacencies to be established.

Before you can enable EIGRP route authentication, you must enable EIGRP.

To enable EIGRP authentication on an interface, perform the following steps:

- 
- Step 1** Enter interface configuration mode for the interface on which you are configuring EIGRP message authentication by entering the following command:

```
hostname(config)# interface phy_if
```

- Step 2** Enable MD5 authentication of EIGRP packets by entering the following command:

```
hostname(config-if)# authentication mode eigrp as-num md5
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the FWSM. If EIGRP is not enabled or if you enter the wrong number, the FWSM returns the following error message:

```
% System(100) specified does not exist
```

- Step 3** Configure the key used by the MD5 algorithm by entering the following command:

```
hostname(config-if)# authentication key eigrp as-num key key-id key-id
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the FWSM. If EIGRP is not enabled or if you enter the wrong number, the FWSM returns the following error message:

```
% System(100) specified does not exist
```

The *key* argument can contain up to 16 characters. The *key-id* argument is a number from 0 to 255.

---

## Defining an EIGRP Neighbor

EIGRP hello packets are sent as multicast packets. If an EIGRP neighbor is located across a nonbroadcast network, such as a tunnel, you must manually define that neighbor. When you manually define an EIGRP neighbor, hello packets are sent to that neighbor as unicast messages.

To manually define an EIGRP neighbor, perform the following steps:

- 
- Step 1** Enter router configuration mode for the EIGRP routing process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

- Step 2** Define the static neighbor by entering the following command:

```
hostname(config-router)# neighbor ip-addr interface if_name
```

The *ip-addr* argument is the IP address of the neighbor. The *if-name* argument is the name of the interface, as specified by the **nameif** command, through which that neighbor is available. You can define multiple neighbors for an EIGRP routing process.

---

## Redistributing Routes Into EIGRP

You can redistribute routes discovered by OSPF into the EIGRP routing process. You can also redistribute static and connected routes into the EIGRP routing process. You do not need to redistribute static or connected routes if they fall within the range of a **network** statement in the EIGRP configuration.

To redistribute routes into the EIGRP routing process, perform the following steps:

- 
- Step 1** (Optional) Create a route-map to further define which routes from the specified routing protocol are redistributed in to the RIP routing process. See the “[Defining a Route Map](#)” section on page 8-5 for more information about creating a route map.

- Step 2** Enter router configuration mode for the EIGRP routing process:

```
hostname(config)# router eigrp as-num
```

- Step 3** (Optional) Specify the default metrics that should be applied to routes redistributed into the EIGRP routing process by entering the following command:

```
hostname(config-router)# default-metric bandwidth delay reliability loading mtu
```

If you do not specify a **default-metric** in the EIGRP router configuration, you must specify the metric values in each **redistribute** command. If you specify the EIGRP metrics in the **redistribute** command and have the **default-metric** command in the EIGRP router configuration, the metrics in the **redistribute** command are used.

- Step 4** Choose one of the following options to redistribute the selected route type into the EIGRP routing process.

- To redistribute connected routes into the EIGRP routing process, enter the following command:

```
hostname(config-router): redistribute connected [metric bandwidth delay reliability  
loading mtu] [route-map map_name]
```

- To redistribute static routes into the EIGRP routing process, enter the following command:

```
hostname(config-router): redistribute static [metric bandwidth delay reliability  
loading mtu] [route-map map_name]
```

- To redistribute routes from an OSPF routing process into the EIGRP routing process, enter the following command:

```
hostname(config-router): redistribute ospf pid [match {internal | external [1 | 2] |  
nssa-external [1 | 2]}] [metric bandwidth delay reliability loading mtu] [route-map  
map_name]
```

You must specify the EIGRP metric values in the **redistribute** command if you do not have a **default-metric** command in the EIGRP router configuration.

## Configuring the EIGRP Hello Interval and Hold Time

The FWSM periodically sends hello packets to discover neighbors and to learn when neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

The hello packet advertises the FWSM hold time. The hold time indicates to EIGRP neighbors the length of time the neighbor should consider the FWSM reachable. If the neighbor does not receive a hello packet within the advertised hold time, then the FWSM is considered unreachable. By default, the advertised hold time is 15 seconds (three times the hello interval).

Both the hello interval and the advertised hold time are configured on a per-interface basis. We recommend setting the hold time to be at minimum three times the hello interval.

To configure the hello interval and advertised hold time, perform the following steps:

- 
- Step 1** Enter interface configuration mode for the interface on which you are configuring hello interval or advertised hold time by entering the following command:

```
hostname(config)# interface phy_if
```

- Step 2** To change the hello interval, enter the following command:

```
hostname(config)# hello-interval eigrp as-num seconds
```

- Step 3** To change the hold time, enter the following command:

```
hostname(config)# hold-time eigrp as-num seconds
```

---

## Disabling Automatic Route Summarization

Automatic route summarization is enabled by default. The EIGRP routing process summarizes on network number boundaries. This can cause routing problems if you have non-contiguous networks.

For example, if you have a router with the networks 192.168.1.0, 192.168.2.0, and 192.168.3.0 connected to it, and those networks all participate in EIGRP, the EIGRP routing process creates the summary address 192.168.0.0 for those routes. If an additional router is added to the network with the networks 192.168.10.0 and 192.168.11.0, and those networks participate in EIGRP, they will also be summarized as 192.168.0.0. To prevent the possibility of traffic being routed to the wrong location, you should disable automatic route summarization on the routers creating the conflicting summary addresses.

To disable automatic router summarization, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no auto-summary
```

**Note**

Automatic summary addresses have an administrative distance of 5. You cannot configure this value.

## Configuring Summary Aggregate Addresses

You can configure a summary addresses on a per-interface basis. You need to manually define summary addresses if you want to create summary addresses that do not occur at a network number boundary or if you want to use summary addresses on a FWSM with automatic route summarization disabled. If any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

To create a summary address, perform the following steps:

- 
- Step 1** Enter interface configuration mode for the interface on which you are creating a summary address by entering the following command:

```
hostname(config)# interface phy_if
```

- Step 2** Create the summary address by entering the following command:

```
hostname(config-if)# summary-address eigrp as-num address mask [distance]
```

By default, EIGRP summary addresses that you define have an administrative distance of 5. You can change this value by specifying the optional *distance* argument in the **summary-address** command.

---

## Disabling EIGRP Split Horizon

Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks, there may be situations where this behavior is not desired. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

If you disable split horizon on an interface, you must disable it for all routers and access servers on that interface.

To disable EIGRP split-horizon, perform the following steps:

- 
- Step 1** Enter interface configuration mode for the interface on which you are disabling split horizon by entering the following command:

```
hostname(config)# interface phy_if
```

**Step 2** To disable split horizon, enter the following command:

```
hostname(config-if)# no split-horizon eigrp as-number
```

---

## Changing the Interface Delay Value

The interface delay value is used in EIGRP distance calculations. You can modify this value on a per-interface basis.

To change the delay value, perform the following steps:

**Step 1** Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP by entering the following command:

```
hostname(config)# interface phy_if
```

**Step 2** To change the delay value, enter the following command:

```
hostname(config-if)# delay value
```

The *value* entered is in tens of microseconds. So, to set the delay for 2000 microseconds, you would enter a *value* of 200.

**Step 3** (Optional) To view the delay value assigned to an interface, use the **show interface** command.

---

## Monitoring EIGRP

You can use the following commands to monitor the EIGRP routing process. For examples and descriptions of the command output, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

- To display the EIGRP event log, enter the following command:

```
hostname# show eigrp [as-number] events [{start end} | type]
```

- To display the interfaces participating in EIGRP routing, enter the following command:

```
hostname# show eigrp [as-number] interfaces [if-name] [detail]
```

- To display the EIGRP neighbor table, enter the following command:

```
hostname# show eigrp [as-number] neighbors [detail | static] [if-name]
```

- To display the EIGRP topology table, enter the following command:

```
hostname# show eigrp [as-number] topology [ip-addr [mask] | active | all-links | pending | summary | zero-successors]
```

- To display EIGRP traffic statistics, enter the following command:

```
hostname# show eigrp [as-number] traffic
```

## Disabling Neighbor Change and Warning Message Logging

By default neighbor change, and neighbor warning messages are logged. You can disable the logging of neighbor change message and neighbor warning messages.

- To disable the logging of neighbor change messages, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no eigrp log-neighbor-changes
```

- To disable the logging of neighbor warning messages, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no eigrp log-neighbor-warnings
```

## Configuring Asymmetric Routing Support

In some situations, return traffic for a session may be routed through a different interface than it originated from. In failover configurations, return traffic for a connection that originated on one unit may return through the peer unit. This most commonly occurs when two interfaces on a single FWSM, or two FWSMs in a failover pair, are connected to different service providers and the outbound connection does not use a NAT address. By default, the FWSM drops the return traffic because there is no connection information for the traffic.

You can prevent the return traffic from being dropped using the **asr-group** command on interfaces where this is likely to occur. When an interface configured with the **asr-group** command receives a packet for which it has no session information, it checks the session information for the other interfaces that are in the same group.



### Note

In failover configurations, you must enable Stateful Failover for session information to be passed from the standby unit or failover group to the active unit or failover group.

If it does not find a match, the packet is dropped. If it finds a match, then one of the following actions occurs:

- If the incoming traffic originated on a peer unit in a failover configuration, some or all of the layer 2 header is rewritten and the packet is redirected to the other unit. This redirection continues as long as the session is active.
- If the incoming traffic originated on a different interface on the same unit, some or all of the layer 2 header is rewritten and the packet is re-injected into the stream.

This section contains the following topics:

- [Adding Interfaces to ASR Groups, page 8-31](#)
- [Asymmetric Routing Support Example, page 8-31](#)

## Adding Interfaces to ASR Groups

Enter the following commands to add an interface to an asymmetric routing group. Stateful Failover must be enabled for asymmetric routing support to function properly between units in failover configurations.

```
hostname/ctx1(config)# interface if
hostname/ctx1(config-if)# asr-group num
```

Valid values for *num* range from 1 to 32. You need to enter the command for each interface that will participate in the ASR group. You can create up to 32 ASR groups and assign a maximum of 8 interfaces to each group.



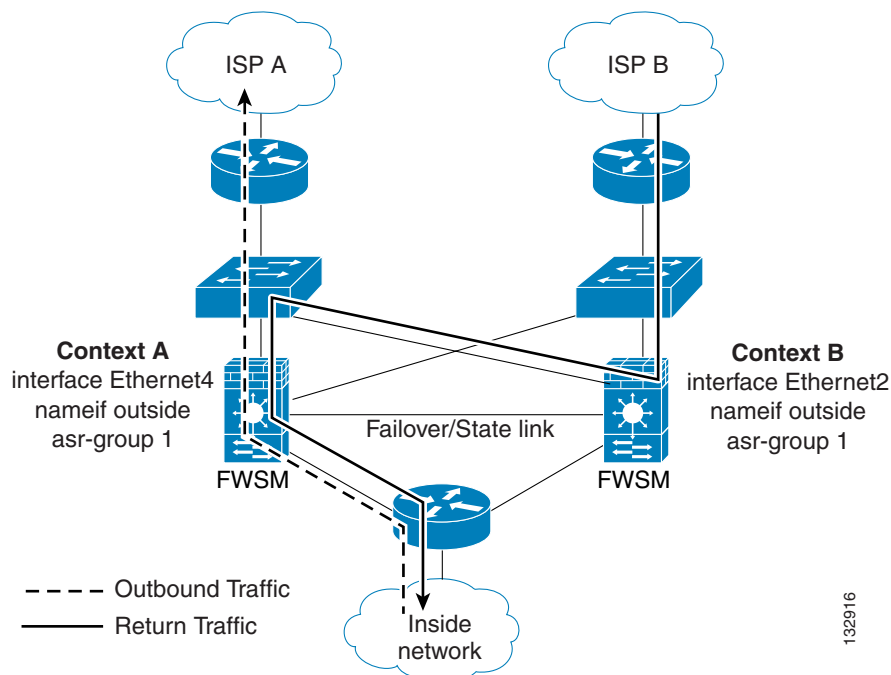
### Note

The upstream and downstream routers must use one MAC address per VLAN and have different MAC addresses for different VLANs to allow for the redirection of packets from a standby unit to an active unit in failover configurations.

## Asymmetric Routing Support Example

Figure 8-1 shows an example of using the **asr-group** command for asymmetric routing support in an Active/Active failover configuration.

**Figure 8-1 ASR Example with Active/Active Failover**



Context A is active on one unit and context B is active on the other. Each context has an interface named “outside”, both of which are configured as part of **asr-group 1**. The outbound traffic is routed through the unit where context A is active. However, the return traffic is being routed through the unit where context B is active. Normally, the return traffic would be dropped because there is no session information

for the traffic on the unit. However, because the interface is configured with an **asr-group** number, the unit looks at the session information for any other interfaces with the same **asr-group** assigned to it. It finds the session information in the outside interface for context A, which is in the standby state on the unit, and forwards the return traffic to the unit where context A is active.

The traffic is forwarded though the outside interface of context A on the unit where context A is in the standby state and returns through the outside interface of context A on the unit where context A is in the active state. This forwarding continues as needed until the session ends.

## Configuring Route Health Injection



### Note

This feature depends on Cisco IOS Release 12.2(33)SXI or later, and is only available on the Catalyst 6500 switch.

Route Health Injection, or RHI, is used for injecting the connected routes, static routes, and NAT addresses configured on the FWSM into the MSFC routing table. In multiple context mode, this feature is especially valuable because of the lack of dynamic routing protocol support. The MSFC can then redistribute the route to other routing tables.

This section includes information on the following topics:

- [Route Health Injection Overview, page 8-32](#)
- [RHI Guidelines, page 8-33](#)
- [Enabling RHI, page 8-33](#)

## Route Health Injection Overview

For connected routes, static routes, and NAT addresses, the FWSM can inject routes into the routing table of the switch; these routes specify the IP address of the FWSM interface as the next hop IP address for each of these FWSM networks.

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

Because the FWSM only supports OSPF or other dynamic routing protocols in single context mode, RHI can be used in multiple mode to inject routes to the MSFC, which can then redistribute these routes through OSPF or other dynamic routing protocols. This allows the FWSM to redistribute FWSM routes through OSPF or other dynamic routing protocols even when running in multiple mode, by utilizing the MSFC routing protocols and RHI.

In a failover scenario, RHI routes are injected from only Active FWSM (applicable in both Active/Standby and Active/Active scenario). If you have FWSM failover between two chassis in Active/Active failover mode, both of the FWSM networks inject routes to their corresponding MSFC, corresponding to the contexts that is in the Active state.

Additionally, if you have HSRP configured between two MSFCs on other interfaces which receive traffic targeted towards either of the two FWSMs, you must choose a routing protocol configured between the two MSFCs. This ensures that each MFSC knows the routes that can be reached through the other FWSM that is not in the same chassis. If there is no exchange of routing information between the two MSFCs, information will not be received and the system will not respond due to the fact that the HSRP Active

MFSC may receive a packet targeted towards a network that can be reached thru FWSM in the other chassis. In that case, the HSRP Active MSFC did not learn of this route from the other MSFC, it may drop the packet (or) incorrectly forwards it to it's default gateway.

The FWSM injects routes into the MSFC using SCP messages.

## RHI Guidelines

- RHI is supported in both single and multiple context mode.
- RHI is supported in routed firewall mode; it is not supported in transparent mode.
- RHI is supported with failover (Active/Standby and Active/Active).
- The FWSM interface that you specify as the next hop interface must be an SVI between the FWSM and the MSFC. See the “[Adding Switched Virtual Interfaces to the MSFC](#)” section on page 2-4.

## Enabling RHI

To configure RHI, perform the following steps:

**Step 1** (Optional) If you want to limit the routes that you inject for each type (connected, static, and NAT), you can limit the routes you want to inject to those that match one of the following objects:

- **route-map**—See the “[Defining a Route Map](#)” section on page 8-5. Route maps are only available in single context mode.
- **access-list standard**—See the “[Adding a Standard Access List](#)” section on page 12-11.
- (NAT only) **global**—See the “[Configuring Dynamic NAT or PAT](#)” section on page 15-25.

**Step 2** Enable RHI by entering the following command:

```
hostname(config)# route-inject
```

The CLI enters route-inject configuration mode. You can only configure one **route-inject** command.

**Step 3** To inject NAT address routes, enter the following command:

```
hostname(config-route-inject)# redistribute nat [route-map map_name | access-list acl_id | global-pool pool_id] interface interface_name
```

where the **interface** *interface\_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all mapped addresses that you define in **static** and **global** commands are injected.

If you want to limit the NAT addresses injected, you can specify the **route-map**, **access-list**, or **global-pool** argument; only matching addresses are injected. For the **global-pool** argument, make sure the **global** command NAT ID that you specify is on the same interface as the **redistribute** command. If you use the same NAT ID for multiple **global** commands on multiple interfaces, only those commands on the matching interface as the **redistribute** command are used.

You can enter only one **redistribute nat** command.

**Step 4** To inject connected routes, enter the following command:

```
hostname(config-route-inject)# redistribute connected [route-map map_name | access-list acl_id] interface interface_name
```

where the **interface** *interface\_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all connected routes are injected.

If you want to limit the routes injected, you can specify the **route-map** or **access-list** argument; only matching addresses are injected.

You can enter only one **redistribute connected** command.

**Step 5** To inject static routes, enter the following command:

```
hostname(config-route-inject)# redistribute static [route-map map_name |
access-list acl_id] interface interface_name
```

where the **interface** *interface\_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all static routes are injected.

If you want to limit the routes injected, you can specify the **route-map** or **access-list** argument; only matching addresses are injected.

You can enter only one **redistribute static** command.

The following example injects NAT addresses that match access list **acl1**; 209.165.201.0/30 is injected with a nexthop of 209.165.200.225 (the active IP address of the outside interface) on VLAN 20. The 209.165.201.10 through .16 addresses are not injected.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.225 255.255.255.224 standby 209.165.200.226
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 209.165.201.0 255.255.255.252
hostname(config)# global (outside) 10 209.165.201.1-209.165.201.2 netmask 255.255.255.0
hostname(config)# global (outside) 10 209.165.201.10-209.165.201.16 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat access-list acl1 interface outside
```

The following example injects 209.165.202.129 through .131 and 209.165.202.140 through .146 with a nexthop 209.165.200.250 on VLAN 20. The global pools on the dmz interface, and the global pool 20 on the outside interface are not included.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.250 255.255.255.224 standby 209.165.200.251
hostname(config-if)# exit
hostname(config)# global (dmz) 10 209.165.201.1-209.165.201.10 netmask 255.255.255.0
hostname(config)# global (outside) 10 209.165.202.129-209.165.202.131 netmask
255.255.255.0
hostname(config)# global (outside) 10 209.165.202.140-209.165.202.146 netmask
255.255.255.0
hostname(config)# global (outside) 20 209.165.202.150-209.165.202.155 netmask
255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat global-pool 10 interface outside
```

The following example injects 209.165.201.0/27 and 192.0.2.0/24 with a nexthop of 209.165.200.225 on VLAN 20. 209.165.202.128/27 is not injected.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.225 255.255.255.224 standby 209.165.200.226
```

```
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 209.165.201.0 255.255.255.224
hostname(config)# access-list acl2 standard permit 192.0.2.0 255.255.255.0
hostname(config)# route-map map1 permit 10
hostname(config-route-map)# match ip address acl1 acl2
hostname(config-route-map)# exit
hostname(config)# route inside 209.165.201.0 255.255.255.224 10.1.1.1
hostname(config)# route inside 192.0.2.0 255.255.255.0 10.1.1.1
hostname(config)# route inside 209.165.202.128 255.255.255.224 10.1.1.1
hostname(config)# route-inject
hostname(config-route-inject)# redistribute static route-map map1 interface outside
```

## Configuring DHCP

DHCP provides network configuration parameters, such as IP addresses, to DHCP clients. The FWSM can provide a DHCP server or DHCP relay services to DHCP clients attached to FWSM interfaces. The DHCP server provides network configuration parameters directly to DHCP clients. DHCP relay passes DHCP requests received on one interface to an external DHCP server located behind a different interface.

This section includes the following topics:

- [Configuring a DHCP Server, page 8-35](#)
- [Configuring DHCP Relay Services, page 8-39](#)

## Configuring a DHCP Server

This section describes how to configure DHCP server provided by the FWSM. This section includes the following topics:

- [Enabling the DHCP Server, page 8-35](#)
- [Configuring DHCP Options, page 8-37](#)
- [Using Cisco IP Phones with a DHCP Server, page 8-38](#)

## Enabling the DHCP Server

The FWSM can act as a DHCP server. DHCP is a protocol that supplies network settings to hosts including the host IP address, the default gateway, and a DNS server.



### Note

The FWSM DHCP server does not support BOOTP requests.

In multiple context mode, you cannot enable the DHCP server or DHCP relay on an interface that is used by more than one context.

You can configure a DHCP server on each interface of the FWSM. Each interface can have its own pool of addresses to draw from. However the other DHCP settings, such as DNS servers, domain name, options, ping timeout, and WINS servers, are configured globally and used by the DHCP server on all interfaces.

**Note**

You can add up to four DHCP relay servers per interface; however, there is a limit of ten DHCP relay servers total that can be configured on the FWSM. You must add at least one **dhcprelay server** command to the FWSM configuration before you can enter the **dhcprelay enable** command. You cannot configure a DHCP client on an interface that has a DHCP relay server configured.

You cannot configure a DHCP client or DHCP Relay services on an interface on which the server is enabled. Additionally, DHCP clients must be directly connected to the interface on which the server is enabled.

To enable the DHCP server on a given FWSM interface, perform the following steps:

- Step 1** Create a DHCP address pool. Enter the following command to define the address pool:

```
hostname(config)# dhcpd address ip_address-ip_address interface_name
```

The FWSM assigns a client one of the addresses from this pool to use for a given length of time. These addresses are the local, untranslated addresses for the directly connected network.

The address pool must be on the same subnet as the FWSM interface.

- Step 2** (Optional) To specify the IP address(es) of the DNS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd dns dns1 [dns2]
```

You can specify up to two DNS servers.

- Step 3** (Optional) To specify the IP address(es) of the WINS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd wins wins1 [wins2]
```

You can specify up to two WINS servers.

- Step 4** (Optional) To change the lease length to be granted to the client, enter the following command:

```
hostname(config)# dhcpd lease lease_length
```

This lease equals the amount of time (in seconds) the client can use its allocated IP address before the lease expires. Enter a value between 0 to 1,048,575. The default value is 3600 seconds.

- Step 5** (Optional) To configure the domain name the client uses, enter the following command:

```
hostname(config)# dhcpd domain domain_name
```

- Step 6** (Optional) To configure the DHCP ping timeout value, enter the following command:

```
hostname(config)# dhcpd ping_timeout milliseconds
```

To avoid address conflicts, the FWSM sends two ICMP ping packets to an address before assigning that address to a DHCP client. This command specifies the timeout value for those packets.

- Step 7** (Transparent Firewall Mode) Define a default gateway. To define the default gateway that is sent to DHCP clients, enter the following command:

```
hostname(config)# dhcpd option 3 ip gateway_ip
```

If you do not use the DHCP option 3 to define the default gateway, DHCP clients use the IP address of the management interface. The management interface does not route traffic.

- Step 8** To enable the DHCP daemon within the FWSM to listen for DHCP client requests on the enabled interface, enter the following command:

```
hostname(config)# dhcpd enable interface_name
```

For example, to assign the range 10.0.1.101 to 10.0.1.110 to hosts connected to the inside interface, enter the following commands:

```
hostname(config)# dhcpd address 10.0.1.101-10.0.1.110 inside
hostname(config)# dhcpd dns 209.165.201.2 209.165.202.129
hostname(config)# dhcpd wins 209.165.201.5
hostname(config)# dhcpd lease 3000
hostname(config)# dhcpd domain example.com
hostname(config)# dhcpd enable inside
```

## Configuring DHCP Options

You can configure the FWSM to send information for the DHCP options listed in RFC 2132. The DHCP options fall into one of three categories:

- Options that return an IP address.
- Options that return a text string.
- Options that return a hexadecimal value.

The FWSM supports all three categories of DHCP options. To configure a DHCP option, do one of the following:

- To configure a DHCP option that returns one or two IP addresses, enter the following command:

```
hostname(config)# dhcpd option code ip addr_1 [addr_2]
```

- To configure a DHCP option that returns a text string, enter the following command:

```
hostname(config)# dhcpd option code ascii text
```

- To configure a DHCP option that returns a hexadecimal value, enter the following command:

```
hostname(config)# dhcpd option code hex value
```



### Note

The FWSM does not verify that the option type and value that you provide match the expected type and value for the option code as defined in RFC 2132. For example, you can enter **dhcpd option 46 ascii hello**, and the FWSM accepts the configuration although option 46 is defined in RFC 2132 as expecting a single-digit, hexadecimal value. For more information about the option codes and their associated types and expected values, refer to RFC 2132.

Table 8-1 shows the DHCP options that are not supported by the **dhcpd option** command:

**Table 8-1**      *Unsupported DHCP Options*

Option Code	Description
0	DHCPOPT_PAD
1	HCPOPT_SUBNET_MASK
12	DHCPOPT_HOST_NAME
50	DHCPOPT_REQUESTED_ADDRESS

**Table 8-1** *Unsupported DHCP Options*

Option Code	Description
51	DHCPOPT_LEASE_TIME
52	DHCPOPT_OPTION_OVERLOAD
53	DHCPOPT_MESSAGE_TYPE
54	DHCPOPT_SERVER_IDENTIFIER
58	DHCPOPT_RENEWAL_TIME
59	DHCPOPT_REBINDING_TIME
61	DHCPOPT_CLIENT_IDENTIFIER
67	DHCPOPT_BOOT_FILE_NAME
82	DHCPOPT_RELAY_INFORMATION
255	DHCPOPT_END

Specific options, DHCP option 3, 66, and 150, are used to configure Cisco IP Phones. See the [“Using Cisco IP Phones with a DHCP Server”](#) section on page 8-38 topic for more information about configuring those options.

## Using Cisco IP Phones with a DHCP Server

Enterprises with small branch offices that implement a Cisco IP Telephony Voice over IP solution typically implement Cisco CallManager at a central office to control Cisco IP Phones at small branch offices. This implementation allows centralized call processing, reduces the equipment required, and eliminates the administration of additional Cisco CallManager and other servers at branch offices.

Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address preconfigured, it sends a request with option 150 or 66 to the DHCP server to obtain this information.

- DHCP option 150 provides the IP addresses of a list of TFTP servers.
- DHCP option 66 gives the IP address or the hostname of a single TFTP server.

Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route.

Cisco IP Phones might include both option 150 and 66 in a single request. In this case, the FWSM DHCP server provides values for both options in the response if they are configured on the FWSM.

You can configure the FWSM to send information for most options listed in RFC 2132. The following table shows the syntax for any option number, as well as the syntax for commonly-used options 66, 150, and 3:

- To provide information for DHCP requests that include an option number as specified in RFC 2132, enter the following command:

```
hostname(config)# dhcpd option number value
```

- To provide the IP address or name of a TFTP server for option 66, enter the following command:

```
hostname(config)# dhcpd option 66 ascii server_name
```

- To provide the IP address or names of one or two TFTP servers for option 150, enter the following command:

```
hostname(config)# dhcpd option 150 ip server_ip1 [server_ip2]
```

The *server\_ip1* is the IP address or name of the primary TFTP server while *server\_ip2* is the IP address or name of the secondary TFTP server. A maximum of two TFTP servers can be identified using option 150.

- To provide set the default route, enter the following command:

```
hostname(config)# dhcpd option 3 ip router_ip1
```

## Configuring DHCP Relay Services

This section describes how to configure DHCP relay services provided by the FWSM. This section includes the following topics:

- [DHCP Relay Overview, page 8-39](#)
- [Configuring the DHCP Relay Agent, page 8-39](#)
- [Preserving DHCP Option 82, page 8-41](#)
- [Verifying the DHCP Relay Configuration, page 8-41](#)

### DHCP Relay Overview

You can configure a DHCP relay agent to forward DHCP requests received on an interface to one or more DHCP servers. When a DHCP request enters an interface, the DHCP servers to which the FWSM relays the request depends on your configuration. You can configure the following types of servers:

- Interface-specific DHCP servers—When a request enters a particular interface, then the FWSM relays the request only to the interface-specific servers.
- Global DHCP servers—When a request enters an interface that does not have interface-specific servers configured, then the FWSM relays the request to all global servers. If the interface has interface-specific servers, then the global servers are not used.

The following restrictions apply to the use of the DHCP relay agent:

- The relay agent cannot be enabled if the DHCP server feature is also enabled.
- DHCP Relay services are not available in transparent firewall mode. You can, however, allow DHCP traffic through using an access list. To allow DHCP requests and replies through the FWSM in transparent mode, you need to configure two access lists, one that allows DHCP requests from the inside interface to the outside, and one that allows the replies from the server in the other direction.
- Clients must be directly-connected to the FWSM and cannot send requests through another relay agent or a router.
- For multiple context mode, you cannot enable DHCP relay on an interface that is used by more than one context.

### Configuring the DHCP Relay Agent

To enable DHCP relay, perform the following steps:

---

**Step 1** Set the IP addresses of DHCP servers using one or both of the following methods:

- To configure an interface-specific server, enter the following commands:

```
hostname(config)# interface {vlan vlan_id | mapped_name}
```

```
hostname(config-if)# dhcprelay server ip_address
```

Where the **vlan** *vlan\_id* or *mapped\_interface* argument is the interface on which you want to enable DHCP relay.

You can enter the **dhcprelay server** command up to 4 times per interface, with a maximum of 10 servers allowed (including global servers) per context or in single mode.

The interface-specific servers take precedence over any global servers configured.

The DHCP servers cannot reside on the same interface on which you enable DHCP relay. (The FWSM determines which interface is connected to the DHCP server by using the routing table.)



**Note** If you configure an interface-specific server address after a connection has already been set up between a client and an existing global DHCP server, the client keeps using the global server until the server address lease expires. After the lease expires, new connections use the interface-specific server.

- To configure a global server, enter the following command:

```
hostname(config)# dhcprelay server ip_address if_name
```

Where the *if\_name* argument is the interface connected to the DHCP server. The DHCP server must reside on a different interface from the DHCP clients where you enable DHCP relay.

You can use this command up to 10 times to identify up to 10 servers, including any interface-specific servers.

**Step 2** To enable DHCP relay on the interface connected to the clients, enter the following command:

```
hostname(config)# dhcprelay enable interface
```

You can enable DHCP relay on multiple interfaces; however, you cannot configure DHCP relay on any interfaces that are connected to the DHCP servers. For example, you can configure DHCP relay on *inside1* and *inside2* interfaces, and configure DHCP servers on *outside* and *dmz* interfaces. You cannot configure any servers on *inside1* or *inside2*.

**Step 3** (Optional) To set the number of seconds allowed for relay address negotiation, enter the following command:

```
hostname(config)# dhcprelay timeout seconds
```

**Step 4** (Optional) To change the first default router address in the packet sent from the DHCP server to the address of the FWSM interface, enter the following command:

```
hostname(config)# dhcprelay setroute if_name
```

This action allows the client to set its default route to point to the FWSM even if the DHCP server specifies a different router.

If there is no default router option in the packet, the FWSM adds one containing the interface address.

The following example enables the FWSM to forward DHCP requests from clients connected to the *inside1* and *inside2* interfaces to a global DHCP server on the *outside* interface and a global DHCP server on the *DMZ* interface:

```
hostname(config)# dhcprelay server 209.165.200.225 outside
hostname(config)# dhcprelay server 209.165.201.4 dmz
hostname(config)# dhcprelay enable inside1
hostname(config)# dhcprelay setroute inside1
```

```
hostname(config)# dhcprelay enable inside2  
hostname(config)# dhcprelay setroute inside2
```

The following example enables the FWSM to forward DHCP requests from clients connected to the inside1 interface (on vlan 20) to an interface-specific DHCP server (on the outside interface). The inside2 interface uses the global DHCP servers on the outside and DMZ interfaces. Note that the global DHCP server on the outside interface is the same as the interface-specific server for inside1.

```
hostname(config)# interface vlan 20  
hostname(config-if)# dhcprelay server 209.165.200.225  
hostname(config)# dhcprelay server 209.165.200.225 outside  
hostname(config)# dhcprelay server 209.165.201.4 dmz  
hostname(config)# dhcprelay enable inside1  
hostname(config)# dhcprelay setroute inside1  
hostname(config)# dhcprelay enable inside2  
hostname(config)# dhcprelay setroute inside2
```

## Preserving DHCP Option 82

This section describes the DHCP option 82 feature. This feature enables the DHCP relay agent to include information about itself and the attached client when forwarding DHCP requests from a DHCP client to a DHCP server.

If the DHCP relay agent receives a DHCP packet from untrusted sources with option 82 already set, but the giaddr (the DHCP relay agent address that will be set by the relay agent before it forwards the packet to the server) field set to zero, or when it is behind DSLAM devices, then it will drop that packet by default. You can optionally preserve option 82 and forward the packet by identifying an interface as a trusted interface. This feature makes sure that DHCP snooping and IP source guard features on the switch work along with the FWSM.

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table.

You can enable this feature on interfaces configured with IPv4 and IPv6 addresses.

To configure a particular interface as a trusted interface that preserves option 82, enter the following commands:

```
hostname(config)# interface interface {vlan vlan_id | mapped_name}  
hostname(config-if)# dhcprelay information trusted
```

To configure all interfaces as trusted interfaces, enter the following command:

```
hostname(config)# dhcprelay information trust-all
```

All the interfaces will be set as trusted interfaces except interfaces which are shared and the interface on which the DHCP server is configured.

The interface-specific trusted configuration and global trusted configuration can exist together. For example there are three interfaces A, B and C, and a user configures interface A as trusted using the interface-specific command. Then the user configures the global command also.

Now all the three interfaces A, B, and C are trusted interfaces. If you enter the **no dhcprelay information trust-all** command, then interfaces B and C will become non-trusted interfaces. Interface A will continue to be a trusted interface, since the interface-specific trusted configuration is not removed.

## Verifying the DHCP Relay Configuration

To view the interface-specific DHCP relay configuration, enter the following command:

```
hostname# show running-config dhcprelay interface [vlan vlan_id | mapped_name]
```

To view the global DHCP relay configuration, enter the following command:

```
hostname# show running-config dhcprelay global
```