OSPF

This chapter describes how to configure the ASA to route data, perform authentication, and redistribute routing information using the Open Shortest Path First (OSPF) routing protocol.

The chapter includes the following sections:

Information About OSPF

OSPF is an interior gateway routing protocol that uses link states rather than distance vectors for path selection. OSPF propagates link-state advertisements rather than routing table updates. Because only LSAs are exchanged instead of the entire routing tables, OSPF networks converge more quickly than RIP networks.

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. The ASA 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.

The ASA 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.

You can redistribute routes into an OSPF routing process from another OSPF routing process, a RIP routing process, or from static and connected routes configured on OSPF-enabled interfaces.

The ASA supports the following OSPF features:

- Intra-area, interarea, and external (Type I and Type II) routes.
- Virtual links.
- LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Configuring the ASA as a designated router or a designated backup router. The ASA also can be set up as an ABR.
- Stub areas and not-so-stubby areas.
- Area boundary router Type 3 LSA filtering.

OSPF supports MD5 and clear text neighbor authentication. Authentication should be used with all routing protocols when possible because route redistribution between OSPF and other protocols (such as RIP) can potentially be used by attackers to subvert routing information.

If NAT is used, if OSPF is operating on public and private areas, and if address filtering is required, then you need to run two OSPF processes—one process for the public areas and one for the private areas.

A router that has interfaces in multiple areas is called an Area Border Router (ABR). A router that acts as a gateway to redistribute traffic between routers using OSPF and routers using other routing protocols is called an Autonomous System Boundary Router (ASBR).

An ABR uses LSAs to send information about available routes to other OSPF routers. Using ABR Type 3 LSA filtering, you can have separate private and public areas with the ASA acting as an ABR. Type 3 LSAs (interarea routes) can be filtered from one area to other, which allows you to use NAT and OSPF together without advertising private networks.

Note Only Type 3 LSAs can be filtered. If you configure the ASA as an ASBR in a private network, it will send Type 5 LSAs describing private networks, which will get flooded to the entire AS, including public areas.

If NAT is employed but OSPF is only running in public areas, then routes to public networks can be redistributed inside the private network, either as default or Type 5 AS external LSAs. However, you need to configure static routes for the private networks protected by the ASA. Also, you should not mix public and private networks on the same ASA interface.

You can have two OSPF routing processes, one RIP routing process, and one EIGRP routing process running on the ASA at the same time.

OSPF Support for Fast Hello Packets

The OSPF Support for Fast Hello Packets feature provides a way to configure the sending of hello packets in intervals less than 1 second. Such a configuration would result in faster convergence in an Open Shortest Path First (OSPF) network.

Prerequisites for OSPF Support for Fast Hello Packets

OSPF must be configured in the network already or configured at the same time as the OSPF Support for Fast Hello Packets feature.

Information About OSPF Support for Fast Hello Packets

The following sections describe concepts related to OSPF support for fast hello packets:

- OSPF Hello Interval and Dead Interval
- OSPF Fast Hello Packets
- Benefits of OSPF Fast Hello Packets

OSPF Hello Interval and Dead Interval

OSPF hello packets are packets that an OSPF process sends to its OSPF neighbors to maintain connectivity with those neighbors. The hello packets are sent at a configurable interval (in seconds). The defaults are 10 seconds for an Ethernet link and 30 seconds for a non broadcast link. Hello packets include a list of all neighbors for which a hello packet has been received within the dead interval. The dead interval is also a configurable interval (in seconds), and defaults to four times the value of the hello interval. The value of all hello intervals must be the same within a network. Likewise, the value of all dead intervals must be the same within a network.

These two intervals work together to maintain connectivity by indicating that the link is operational. If a router does not receive a hello packet from a neighbor within the dead interval, it will declare that neighbor to be down.

OSPF Fast Hello Packets

OSPF fast hello packets refer to hello packets being sent at intervals of less than 1 second. To understand fast hello packets, you should already understand the relationship between OSPF hello packets and the dead interval. See the section OSPF Hello Interval and Dead Interval , page 30-3.

OSPF fast hello packets are achieved by using the **ospf dead-interval** command. The dead interval is set to 1 second, and the hello-multiplier value is set to the number of hello packets you want sent during that 1 second, thus providing subsecond or "fast" hello packets.

When fast hello packets are configured on the interface, the hello interval advertised in the hello packets that are sent out this interface is set to 0. The hello interval in the hello packets received over this interface is ignored.

The dead interval must be consistent on a segment, whether it is set to 1 second (for fast hello packets) or set to any other value. The hello multiplier need not be the same for the entire segment as long as at least one hello packet is sent within the dead interval.

Benefits of OSPF Fast Hello Packets

The benefit of the OSPF Fast Hello Packets feature is that your OSPF network will experience faster convergence time than it would without fast hello packets. This feature allows you to detect lost neighbors within 1 second. It is especially useful in LAN segments, where neighbor loss might not be detected by the Open System Interconnection (OSI) physical layer and data-link layer.

Implementation Differences Between OSPFv2 and OSPFv3

OSPFv3 is not backward compatible with OSPFv2. To use OSPF to route both IPv4 and IPv6 traffic, you must run both OSPFv2 and OSPFv3 at the same time. They coexist with each other, but do not interact with each other.

The additional features that OSPFv3 provides include the following:

- Protocol processing per link.
- Removal of addressing semantics.
- Addition of flooding scope.
- Support for multiple instances per link.
- Use of the IPv6 link-local address for neighbor discovery and other features.
- LSAs expressed as prefix and prefix length.
- Addition of two LSA types.
- Handling of unknown LSA types.
- Authentication support using the IPsec ESP standard for OSPFv3 routing protocol traffic, as specified by RFC-4552.

Using Clustering

For more information about dynamic routing and clustering, see Dynamic Routing and Clustering, page 26-9.

For more information about using clustering, see Chapter 9, "ASA Cluster."

Licensing Requirements for OSPF

Model	License Requirement
ASAv	Standard or Premium License.
All other models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

OSPFv2 supports single and multiple context mode.

OSPFv3 supports single mode only.

Firewall Mode Guidelines

OSPF supports routed firewall mode only. OSPF does not support transparent firewall mode.

Failover Guidelines

OSPFv2 and OSPFv3 support Stateful Failover.

IPv6 Guidelines

- OSPFv2 does not support IPv6.
- OSPFv3 supports IPv6.
- OSPFv3 uses IPv6 for authentication.
- The ASA installs OSPFv3 routes into the IPv6 RIB, provided it is the best route.
- OSPFv3 packets can be filtered out using IPv6 ACLs in the **capture** command.

Clustering Guidelines

- OSPFv2 and OSPFv3 support clustering.
- OSPFv3 encryption is not supported. An error message appears if you try to configure OSPFv3 encryption in a clustering environment.
- In the spanned interface mode, dynamic routing is not supported on management-only interfaces.
- In individual interface mode, make sure that you establish the master and slave units as either OSPFv2 or OSPFv3 neighbors.
- When you configure both OSPFv2 and EIGRP, you can use either spanned interface mode or individual interface mode; you cannot use the two modes at the same time.
- In individual interface mode, OSPFv2 adjacencies can only be established between two contexts on a shared interface on the master unit. Configuring static neighbors is supported only on point-to point-links; therefore, only one neighbor statement is allowed on an interface.
- The router ID is optional in the OSPFv2, OSPFv3, and EIGRP router configuration mode. If you do not explicitly set a router ID, then a router ID is automatically generated and set to the highest IPv4 address on any data interface in each of the cluster units.
- If the cluster interface mode has not been configured, then only a single, dotted-decimal IPv4 address is allowed as the router ID, and the **cluster pool** option is disabled.
- If the cluster interface mode is set to a spanned configuration, then only a single, dotted-decimal IPv4 address is allowed as the router ID, and the **cluster pool** option is disabled.
- If the cluster interface mode is set to an individual configuration, then the **cluster pool** option is mandatory, and a single, dotted-decimal IPv4 address is not allowed as the router ID.
- When the cluster interface mode is changed from a spanned to an individual configuration and vice versa without specifying the **check-detail** or **nocheck** options, then the entire configuration including the router ID is removed.

- If any of the dynamic routing protocol router ID configurations are incompatible with the new interface mode, then an error message appears on the console and the interface mode CLI fails. The error message has one line per dynamic routing protocol (OSPFv2, OSPFv3, and EIGRP) and lists the names of each context in which the incompatible configuration occurs.
- If the **nocheck** option is specified for the **cluster interface mode** command, then the interface mode is allowed to change although all the router ID configurations may not be compatible with the new mode.
- When the cluster is enabled, the router ID compatibility checks are repeated. If any incompatibility is detected, then the **cluster enable** command fails. The administrator needs to correct the incompatible router ID configuration before the cluster can be enabled.
- When a unit enters a cluster as a slave, then we recommend that you specify the **nocheck** option for the **cluster interface mode** command to avoid any router ID compatibility check failures. The slave unit still inherits the router configuration from the master unit.
- When a mastership role change occurs in the cluster, the following behavior occurs:
  - In spanned interface mode, the router process is active only on the master unit and is in a suspended state on the slave units. Each cluster unit has the same router ID because the configuration has been synchronized from the master unit. As a result, a neighboring router does not notice any change in the router ID of the cluster during a role change.
  - In individual interface mode, the router process is active on all the individual cluster units. Each cluster unit chooses its own distinct router ID from the configured cluster pool. A mastership role change in the cluster does not change the routing topology in any way.

Additional Guidelines

- OSPFv2 and OSPFv3 support multiple instances on an interface.
- OSPFv3 supports encryption through ESP headers in a non-clustered environment.
- OSPFv3 supports Non-Payload Encryption.

Configuring OSPFv2

This section describes how to enable an OSPFv2 process on the ASA.

After you enable OSPFv2, you need to define a route map. For more information, see Defining a Route Map, page 28-5. Then you generate a default route. For more information, see Configuring Static and Default Routes, page 27-2.

After you have defined a route map for the OSPFv2 process, you can customize it for your particular needs, To learn how to customize the OSPFv2 process on the ASA, see Customizing OSPFv2, page 30-8.

To enable OSPFv2, you need to create an OSPFv2 routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

You can enable up to two OSPFv2 process instances. Each OSPFv2 process has its own associated areas and networks.

To enable OSPFv2, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* argument is an internally used identifier for this routing process and 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. If there is only one OSPF process enabled on the ASA, then that process is selected by default. You cannot change the OSPF process ID when editing an existing area.

Step 2	**network** *ip_address mask* **area** *area_id*
	
	Example:
	`ciscoasa(config)# router ospf 2`
	`ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0`
		Defines the IP addresses on which OSPF runs and the area ID for that interface.
		
		When adding a new area, enter the area ID. You can specify the area ID as either a decimal number or an IP address. Valid decimal values range from 0 to 4294967295. You cannot change the area ID when editing an existing area.

Configuring OSPF Fast Hello Packets

This section describes how to configure OSPF Fast Hello Packets.

Detailed Steps

	Command	Purpose
Step 1	**interface port-channel** *number* **Example:** `ciscoasaciscoasa(config)# interface port-channel 10`	Configures an interface and enters the interface configuration mode. The *number* argument indicates the port-channel interface number.
Step 2	**ospf dead-interval minimal hello-multiplier** *no.of times* **Example:** `ciscoasaciscoasa(config-if)# ospf dead-interval minimal hell0-multiplier 5 ciscoasa`	Sets the interval during which at least one hello packet must be received, or else the neighbor is considered down. The *no. of times* argument indicates the number of hello packets to be sent every second. Valid values are between 3 and 20. In this example, OSPF Support for Fast Hello Packets is enabled by specifying the minimal keyword and the hello-multiplier keyword and value. Because the multiplier is set to 5, five hello packets will be sent every second.

Customizing OSPFv2

This section explains how to customize the OSPFv2 processes and includes the following topics:

Redistributing Routes Into OSPFv2

The ASA can control the redistribution of routes between OSPFv2 routing processes.

Note If you want to redistribute a route by defining which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process, you must first generate a default route. See Configuring Static and Default Routes, page 27-2, and then define a route map according to the Defining a Route Map, page 28-5.

To redistribute static, connected, RIP, or OSPFv2 routes into an OSPFv2 process, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* argument is an internally used identifier for this routing process and 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	Do one of the following to redistribute the selected route type into the OSPF routing process:	

Command	Purpose
redistribute connected [[**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*] **Example:** `ciscoasa(config)# redistribute connected 5 type-1 route-map-practice`	Redistributes connected routes into the OSPF routing process.
redistribute static [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name* **Example:** `ciscoasa(config)# redistribute static 5 type-1 route-map-practice`	Redistributes static routes into the OSPF routing process.

redistribute ospf ***pid*** [**match {internal** | **external** [**1** | **2]** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
 	
Example:
`ciscoasa(config)# route-map 1-to-2 permit`
`ciscoasa(config-route-map)# match metric 1`
`ciscoasa(config-route-map)# set metric 5`
`ciscoasa(config-route-map)# set metric-type type-1`
`ciscoasa(config-route-map)# router ospf 2`
`ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2`
	Allows you to redistribute routes from an OSPF routing process into another OSPF routing process.
	
	You can either use the **match** options in this command to match and set route properties, or you can use a route map. The **subnets** option does not have equivalents in the **route-map** command. If you use both a route map and **match** options in the **redistribute** command, then they must match.
	
	The example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The ASA redistributes these routes as external LSAs with a metric of 5 and a metric type of Type 1.

Command	Purpose

redistribute rip [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
 	
Example:
`ciscoasa(config)# redistribute rip 5`
`ciscoasa(config-route-map)# match metric 1`
`ciscoasa(config-route-map)# set metric 5`
`ciscoasa(config-route-map)# set metric-type type-1`
`ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2`
	Allows you to redistribute routes from a RIP routing process into the OSPF routing process.

redistribute eigrp ***as-num*** [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
 	
Example:
`ciscoasa(config)# redistribute eigrp 2`
`ciscoasa(config-route-map)# match metric 1`
`ciscoasa(config-route-map)# set metric 5`
`ciscoasa(config-route-map)# set metric-type type-1`
`ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2`
	Allows you to redistribute routes from an EIGRP routing process into the OSPF routing process.

Configuring Route Summarization When Redistributing Routes Into OSPFv2

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

Routes that match the specified IP address mask pair can be suppressed. The tag value can be used as a match value for controlling redistribution through route maps.

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

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 1`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* argument is an internally used identifier for this routing process and 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	summary-address *ip_address mask* [**not-advertise**] [tag *tag*]
	
	Example:
	`ciscoasa(config)# router ospf 1`
	`ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0`
		Sets the summary address.
		In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement.

Configuring Route Summarization Between OSPFv2 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 so that they are contiguous, you can configure the area boundary router to advertise a summary route that includes 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:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 1`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* argument 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	area *area-id* range *ip-address mask* [**advertise** | **not-advertise**]
	
	Example:
	`ciscoasa(config)# router ospf 1`
	`ciscoasa(config-rtr)# area 17 range 12.1.0.0 255.255.0.0`
		Sets the address range.
		In this example, the address range is set between OSPF areas.

Configuring OSPFv2 Interface Parameters

You can change some interface-specific OSPFv2 parameters, if necessary. You are not required to change 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**. If you configure any of these parameters, be sure that the configurations for all routers on your network have compatible values.

To configure OSPFv2 interface parameters, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	`router ospf` *`process_id`* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* argument is an internally used identifier for this routing process and 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	`network` *`ip_address mask`* `area` *`area_id`*
	
	Example:
	`ciscoasa(config)# router ospf 2`
	`ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0`
		Defines the IP addresses on which OSPF runs and the area ID for that interface.

Step 3	`interface` *`interface_name`* **Example:** `ciscoasa(config)# interface` *`my_interface`*	Allows you to enter interface configuration mode.
Step 4	Do one of the following to configure optional OSPF interface parameters:	
	ospf authentication [**message-digest** | **null**] **Example:** `ciscoasa(config-interface)# ospf authentication message-digest`	Specifies the authentication type for an interface.
	ospf authentication-key *key* **Example:** `ciscoasa(config-interface)# ospf authentication-key cisco`	Allows you to assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication. The *key* argument 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 ASA 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.

Command	Purpose
ospf cost *cost* **Example:** `ciscoasa(config-interface)# ospf cost 20`	Allows you to explicitly specify the cost of sending a packet on an OSPF interface. The *cost* is an integer from 1 to 65535. In this example, the cost is set to 20.
ospf dead-interval *seconds* **Example:** `ciscoasa(config-interface)# ospf dead-interval 40`	Allows you 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. The value must be the same for all nodes on the network. In this example, the dead interval is set to 40.
ospf hello-interval *seconds* **Example:** `ciscoasa(config-interface)# ospf hello-interval 10`	Allows you to specify the length of time between the hello packets that the ASA sends on an OSPF interface. The value must be the same for all nodes on the network. In this example, the hello interval is set to 10.
ospf message-digest-key *key_id* **md5** *key* **Example:** `ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco`	Enables OSPF MD5 authentication. The following argument values can be set: • *key_id*—An identifier in the range from 1 to 255. • *key*—An 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.
ospf priority *number_value* **Example:** `ciscoasa(config-interface)# ospf priority 20`	Allows you to set the priority to help determine the OSPF designated router for a network. The *number_value* argument ranges from 0 to 255. In this example, the priority number value is set to 20.
ospf retransmit-interval *seconds* **Example:** `ciscoasa(config-interface)# ospf retransmit-interval seconds`	Allows you to specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface. The value for *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 8192 seconds. The default value is 5 seconds. In this example, the retransmit-interval value is set to 15.
ospf transmit-delay *seconds* **Example:** `ciscoasa(config-interface)# ospf transmit-delay 5`	Sets the estimated number of seconds required to send a link-state update packet on an OSPF interface. The *seconds* value ranges from 1 to 8192 seconds. The default value is 1 second. In this example, the transmit-delay is 5 seconds.

Command	Purpose
ospf dead-interval minimal hello-interval *multiplier* **Example:** `ciscoasa(config-if)# ospf dead-interval minimal hello-multiplier 6`	Sets the number of hello packets sent during 1 second. Valid values are integers between 3 and 20. In this example the multiplier is 6.
ospf network point-to-point non-broadcast **Example:** `ciscoasa(config-interface)# ospf network point-to-point non-broadcast`	Specifies the interface as a point-to-point, non-broadcast network. When you designate an interface as point-to-point and non-broadcast, you must manually define the OSPF neighbor; dynamic neighbor discovery is not possible. See Defining Static OSPFv2 Neighbors, page 30-17 for more information. Additionally, you can only define one OSPF neighbor on that interface.

Configuring OSPFv2 Area Parameters

You can configure several OSPF area parameters. These area parameters (shown in the following task list) 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 use the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending a summary link advertisement (LSA Type 3) into the stub area.

To specify OSPFv2 area parameters for your network, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	`router ospf` *process_id* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* argument is an internally used identifier for this routing process and 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	Do one of the following to configure optional OSPF area parameters:	

Command	Purpose
area *area-id* **authentication** **Example:** `ciscoasa(config-rtr)# area 0 authentication`	Enables authentication for an OSPF area.
area *area-id* **authentication message-digest** **Example:** `ciscoasa(config-rtr)# area 0 authentication message-digest`	Enables MD5 authentication for an OSPF area.

Configuring an OSPFv2 NSSA

The OSPFv2 implementation of an NSSA is similar to an OSPFv2 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 OSPFv2 to a remote site that is using a different routing protocol with NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPFv2 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 OSPFv2 to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

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 cannot communicate with each other.

To specify area parameters for your network to configure an OSPFv2 NSSA, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF routing process that you want to redistribute. The *process_id* argument 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	Do one of the following to configure optional OSPF NSSA parameters:	
	area *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] **Example:** `ciscoasa(config-rtr)# area 0 nssa`	Defines an NSSA area.
	summary-address *ip_address mask* [**not-advertise**] [tag *tag*] **Example:** `ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0`	Sets the summary address and 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. In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement.

Note OSPF does not support summary-address 0.0.0.0 0.0.0.0.

Configuring an IP Address Pool for Clustering (OSPFv2 and OSPFv3)

You can assign a range of IPv4 addresses for the router ID cluster pool if you are using Individual Interface clustering.

To assign a range of IPv4 addresses for the router ID cluster pool in Individual Interface clustering for OSPFv2 and OSPFv3, enter the following command:

Command	Purpose
**`router-id cluster-pool hostname	A.B.C.D`** *`ip_pool`*  **Example:** `hostname(config)# ip local pool rpool 1.1.1.1-1.1.1.4` `hostname(config)# router ospf 1` `hostname(config-rtr)# router-id cluster-pool rpool` `hostname(config-rtr)# network 17.5.0.0 255.255.0.0 area 1` `hostname(config-rtr)# log-adj-changes`	Specifies the router ID cluster pool for Individual Interface clustering. The **cluster-pool** keyword enables configuration of an IP address pool when Individual Interface clustering is configured. The **hostname** | **A.B.C.D.** keyword specifies the OSPF router ID for this OSPF process. The *ip_pool* argument specifies the name of the IP address pool. **Note** If you are using clustering, then you do not need to specify an IP address pool for the router ID. If you do not configure an IP address pool, then the ASA uses the automatically generated router ID.

:

Defining Static OSPFv2 Neighbors

You need to define static OSPFv2 neighbors to advertise OSPFv2 routes over a point-to-point, non-broadcast network. This feature lets you broadcast OSPFv2 advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

Before you begin, you must create a static route to the OSPFv2 neighbor. See Chapter 27, “Static and Default Routes,” for more information about creating static routes.

To define a static OSPFv2 neighbor, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**`router ospf`** *`process_id`* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPFv2 routing process and enters router configuration mode for this OSPFv2 process. The *process_id* argument is an internally used identifier for this routing process and 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	**`neighbor`** *`addr`* `[`**`interface`** *`if_name`*`]` **Example:** `ciscoasa(config-rtr)# neighbor 255.255.0.0 [interface my_interface]`	Defines the OSPFv2 neighborhood. The *addr* argument is the IP address of the OSPFv2 neighbor. The *if_name* argument is the interface used to communicate with the neighbor. If the OSPF v2neighbor is not on the same network as any of the directly connected interfaces, you must specify the interface.

Configuring Route Calculation Timers

You can configure the delay time between when OSPFv2 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:

Detailed Steps

	Command	Purpose
Step 1	**`router ospf`** *`process_id`* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPFv2 routing process and enters router configuration mode for this OSPFv2 process. The *process_id* argument is an internally used identifier for this routing process and 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	**`timers throttle spf`** *`spf-start spf-hold spf-maximum`* **Example:** `ciscoasa(config-router)# timers throttle spf 500 500 600`	Configures the route calculation times. The *spf-start* argument is the delay time (in milliseconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 600000. The *spf-hold* argument is the minimum time (in milliseconds) between two consecutive SPF calculations. It can be an integer from 0 to 600000. The *spf-maximum* argument is the maximum time (in milliseconds) between two consecutive SPF calculations. It can be integer from 0 to 600000.

Logging Neighbors Going Up or Down

By default, a syslog message is generated when an OSPFv2 neighbor goes up or down.

Configure the **log-adj-changes** command if you want to know about OSPFv2 neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** command provides a higher level view of the peer relationship with less output. Configure the **log-adj-changes detail** command if you want to see messages for each state change.

To log OSPFv2 neighbors going up or down, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config)# router ospf 2`	Creates an OSPFv2 routing process and enters router configuration mode for this OSPFv2 process. The *process_id* argument is an internally used identifier for this routing process and 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	**log-adj-changes** [**detail**] **Example:** `ciscoasa(config-rtr)# log-adj-changes [detail]`	Configures logging for neighbors going up or down.

Configuring OSPFv3

This section describes how to configure OSPFv3 routing processes and includes the following topics:

- Configuring an IPv6 Summary Prefix, page 30-40
- Redistributing IPv6 Routes, page 30-41

Enabling OSPFv3

To enable OSPFv3, you need to create an OSPFv3 routing process, create an area for OSPFv3, enable an interface for OSPFv3, then redistribute the route into the targeted OSPFv3 routing processes.

To enable OSPFv3,enter the following command or perform the following steps:

Command	Purpose
ipv6 router ospf *process-id* **Example:** `ciscoasa(config)# ipv6 router ospf 10`	Creates an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used tag for this routing process and can be any positive integer. This tag does not have to match the tag on any other device; it is for internal use only. You can use a maximum of two processes.

	Command	Purpose
Step 1	**interface** *interface_name* **Example:** `ciscoasa(config)# interface Gigabitethernet0/0`	Enables an interface.
Step 2	**ipv6 ospf** *process-id* **area** *area_id* **Example:** `ciscoasa(config)# ipv6 ospf 200 area 100`	Creates the OSPFv3 routing process with the specified process ID and an area for OSPFv3 with the specified area ID.

Configuring OSPFv3 Interface Parameters

You can change certain interface-specific OSPFv3 parameters, if necessary. You are not required to change any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ipv6 ospf hello-interval** and **ipv6 ospf dead-interval**. If you configure any of these parameters, be sure that the configurations for all routers on your network have compatible values.

To configure OSPFv3 interface parameters for IPv6, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 10`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used tag for this routing process and can be any positive integer. This tag does not have to match the tag on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2	**ipv6 ospf area** [*area-num*] [**instance**]
	
	Example:
	`ciscoasa(config-if)# interface GigabitEthernet3/2.200`
	` vlan 200`
	` nameif outside`
	` security-level 100`
	` ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31`
	` ipv6 address 3001::1/64 standby 3001::8`
	` ipv6 address 6001::1/64 standby 6001::8`
	` ipv6 enable`
	` ospf priority 255`
	` ipv6 ospf cost 100`
	` ipv6 ospf 100 area 10 instance 200`
		Creates an OSPFv3 area.
		
		The *area-num* argument is the area for which authentication is to be enabled and can be either a decimal value or an IP address. The **instance** keyword specifies the area instance ID that is to be assigned to an interface. An interface can have only one OSPFv3 area. You can use the same area on multiple interfaces, and each interface can use a different area instance ID.

Step 3	Do one of the following to configure OSPFv3 interface parameters:	

	ipv6 ospf cost *interface-cost*
	
	Example:
	`ciscoasa(config-if)# interface GigabitEthernet3/2.200`
	` vlan 200`
	` nameif outside`
	` security-level 100`
	` ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31`
	` ipv6 address 3001::1/64 standby 3001::8`
	` ipv6 address 6001::1/64 standby 6001::8`
	` ipv6 enable`
	` ospf priority 255`
	` ipv6 ospf cost 100`
	` ipv6 ospf 100 area 10 instance 200`
		Explicitly specifies the cost of sending a packet on an interface. The *interface-cost* argument specifies an unsigned integer value expressed as the link-state metric, which can range in value from 1 to 65535. The default cost is based on the bandwidth.

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>ipv6 ospf database-filter all out

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf database-filter all out</pre></td><td>Filters outgoing LSAs to an OSPFv3 interface. All outgoing LSAs are flooded to the interface by default.</td></tr>
<tr><td>ipv6 ospf dead-interval seconds

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf dead-interval 60</pre></td><td>Sets the time period in seconds for which hello packets must not be seen before neighbors indicate that the router is down. The value must be the same for all nodes on the network and can range from 1 to 65535. The default is four times the interval set by the ipv6 ospf hello-interval command.</td></tr>
</table>

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>

ipv6 ospf encryption {ipsec spi spi esp encryption-algorithm [[key-encryption-type] key] authentication-algorithm [[key-encryption-type] key | null}

Example:

```
ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf encryption ipsec spi 1001 esp
null sha1
123456789A123456789B123456789C123456789D
```

</td><td>

Specifies the encryption type for an interface. The ipsec keyword specifies the IP security protocol. The spi spi keyword-argument pair specifies the security policy index, which must be in the range of 256 to 42949667295 and entered as a decimal. The esp keyword specifies the encapsulating security payload. The encryption-algorithm argument specifies the encryption algorithm to be used with ESP. Valid values include the following:

- aes-cdc—Enables AES-CDC encryption.
- 3des—Enables 3DES encryption.
- des—Enables DES encryption.
- null—Specifies ESP with no encryption.

The key-encryption-type argument can be one of the following two values:

- 0—The key is not encrypted.
- 7—The key is encrypted.

The key argument specifies the number used in the calculation of the message digest. The number is 32 hexadecimal digits (16 bytes) long. The size of the key depends on the encryption algorithm used. Some algorithms, such as AES-CDC, allow you to choose the size of the key. The authentication-algorithm argument specifies the encryption authentication algorithm to be used, which can be one of the following:

- md5—Enables message digest 5 (MD5).
- sha1—Enables SHA-1.

The null keyword overrides area encryption.

Note If OSPFv3 encryption is enabled on an interface and a neighbor is on different area (for example, area 0), and you want the ASA to form adjacencies with that area, you must change the area on the ASA. After you have changed the area on the ASA to 0, there is a delay of two minutes before the OSPFv3 adjacency comes up.

</td></tr>
<tr><td>

ipv6 ospf flood-reduction

Example:

```
ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf flood reduction
```

</td><td>Specifies the flood reduction of LSAs to the interface.</td></tr>
</table>

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>ipv6 ospf hello-interval seconds

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf hello-interval 15</pre></td><td>Specifies the interval in seconds between hello packets sent on the interface. The value must be the same for all nodes on a specific network and can range from 1 to 65535. The default interval is 10 seconds for Ethernet interfaces and 30 seconds for non-broadcast interfaces.</td></tr>
<tr><td>ipv6 ospf mtu-ignore

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf mtu-ignore</pre></td><td>Disables the OSPF MTU mismatch detection when DBD packets are received. OSPF MTU mismatch detection is enabled by default.</td></tr>
<tr><td>ipv6 ospf network {broadcast | point-to-point non-broadcast}

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf network point-to-point
non-broadcast</pre></td><td>Sets the OSPF network type to a type other than the default, which depends on the network type. The point-to-point non-broadcast keyword sets the network type to point-to-point non-broadcast. The broadcast keyword sets the network type to broadcast.</td></tr>
</table>

Command	Purpose

ipv6 ospf priority *number-value*
 	
Example:
`ciscoasa(config-if)# interface GigabitEthernet3/2.200`
` vlan 200`
` nameif outside`
` security-level 100`
` ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31`
` ipv6 address 3001::1/64 standby 3001::8`
` ipv6 address 6001::1/64 standby 6001::8`
` ipv6 enable`
` ospf priority 255`
` ipv6 ospf cost 100`
` ipv6 ospf 100 area 10 instance 200`
`ipv6 ospf priority 4`
	Sets the router priority, which helps determine the designated router for a network. Valid values range from 0 to 255.

ipv6 ospf neighbor *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**]
 	
Example:
`ciscoasa(config-if)# interface GigabitEthernet3/2.200`
` vlan 200`
` nameif outside`
` security-level 100`
` ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31`
` ipv6 address 3001::1/64 standby 3001::8`
` ipv6 address 6001::1/64 standby 6001::8`
` ipv6 enable`
` ospf priority 255`
` ipv6 ospf cost 100`
` ipv6 ospf 100 area 10 instance 200`
`ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01`
	Configures OSPFv3 router interconnections to non-broadcast networks.

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>ipv6 ospf retransmit-interval seconds

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-interval 8</pre></td><td>Specifies the time in seconds between LSA retransmissions for adjacencies that belong to the interface. The time must be greater than the expected round-trip delay between any two routers on the attached network. Valid values range from 1 to 65535 seconds. The default is 5 seconds.</td></tr>
<tr><td>ipv6 ospf transmit-delay seconds

Example:
<pre>ciscoasa(config-if)# interface
GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0
standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-delay 3</pre></td><td>Sets the estimated time in seconds to send a link-state update packet on the interface. Valid values range from 1 to 65535 seconds. The default is 1 second.</td></tr>
</table>

Configuring OSPFv3 Router Parameters

To configure OSPFv3 router parameters for IPv6, perform the following steps:

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td>ipv6 router ospf process-id

Example:
<pre>ciscoasa(config)# ipv6 router ospf 10</pre></td><td>Enables an OSPFv3 routing process and enters IPv6 router configuration mode.

The process-id argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.</td></tr>
<tr><td>Step 2</td><td colspan="2">Do one of the following to configure optional OSPFv3 router parameters:</td></tr>
</table>

Command	Purpose

`area`
 	
Example:
`ciscoasa(config-rtr)# area 10`
	Configures OSPFv3 area parameters. Supported parameters include the area ID as a decimal value from 0 to 4294967295 and the area ID in the IP address format of **A.B.C.D**.

`default` **Example:** `ciscoasa(config-rtr)# default originate`	Sets a command to its default value. The **originate** parameter distributes the default route.
`default-information` **Example:** `ciscoasa(config-rtr)# default-information`	Controls distribution of default information.
`distance` **Example:** `ciscoasa(config-rtr)# distance 200`	Defines the OSPFv3 route administrative distance based on the route type. Supported parameters include the administrative distance with values from 1 to 254 and **ospf** for the OSPFv3 distance.

`exit`
 	
Example:
`ciscoasa(config-rtr)# exit`
	Exits from IPv6 router configuration mode.

`ignore` **Example:** `ciscoasa(config-rtr)# ignore lsa`	Suppresses the sending of syslog messages with the **lsa** parameter when the router receives a link-state advertisement (LSA) for Type 6 Multicast OSPF (MOSPF) packets.

`log-adjacency-changes`
 	
Example:
`ciscoasa(config-rtr)#`
`log-adjacency-changes detail`
	Configures the router to send a syslog message when an OSPFv3 neighbor goes up or down. With the **detail** parameter, all state changes are logged.

`passive-interface` [*interface_name*]
 	
Example:
`ciscoasa(config-rtr)# passive-interface`
`inside`
	Suppresses the sending and receiving of routing updates on an interface. The *interface_name* argument specifies the name of the interface on which the OSPFv3 process is running.

`redistribute` **Example:** `ciscoasa(config-rtr)# redistribute ospf`	Configures the redistribution of routes from one routing domain into another according to the following parameters: • **connected**—Specifies connected routes. • **ospf**—Specifies OSPFv3 routes. • **static**—Specifies static routes.

Command	Purpose
router-id **Example:** `ciscoasa(config-rtr)# router-id 10.1.1.1`	Creates a fixed router ID for a specified process with the following parameters: • **A.B.C.D**—Specifies the OSPF router ID in IP address format. • **cluster-pool**—Configures an IP address pool when Individual Interface clustering is configured. For more information about IP address pools used in clustering, see Configuring an IP Address Pool for Clustering (OSPFv2 and OSPFv3), page 30-17.

summary-prefix
 	
Example:
`ciscoasa(config-if)# ipv6 router ospf 1`
`ciscoasa(config-router)# router-id 192.168.3.3`
`ciscoasa(config-router)# summary-prefix FECO::/24`
`ciscoasa(config-router)# redistribute static`
	Configures IPv6 address summaries with valid values from 0 to 128. The **X:X:X:X::X/** parameter specifies the IPv6 prefix.

timers
 	
Example:
`ciscoasa(config)# ipv6 router ospf 10`
`ciscoasa(config-rtr)# timers throttle spf 6000 12000 14000`
	Adjusts routing timers. The routing timer parameters are the following:
	• **lsa**—Specifies OSPFv3 LSA timers.
	• **pacing**—Specifies OSPFv3 pacing timers.
	• **throttle**—Specifies OSPFv3 throttle timers.

Configuring OSPFv3 Area Parameters

To configure OSPFv3 area parameters, perform the following steps:

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	Do one of the following to configure optional OSPFv3 area parameters:	
	area *area-id* **default-cost** *cost* **Example:** `ciscoasa(config-rtr)# area 1 default-cost nssa`	Sets the summary default cost of an NSSA area or a stub area.

Command	Purpose
area *area-id* **range** *ipv6-prefix/ prefix-length* [**advertise** | **not advertise**] [**cost** *cost*] **Example:** `ciscoasa(config-rtr)# area 1 range FE01:1::1/64`	Summarizes routes that match the address and mask for border routers only. The *area-id* argument identifies the area for which routes are to be summarized. The value can be specified as a decimal or an IPv6 prefix. The *ipv6-prefix* argument specifies the IPv6 prefix. The *prefix-length* argument specifies the prefix length. The **advertise** keyword sets the address range status to advertised and generates a Type 3 summary LSA. The **not-advertise** keyword sets the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and the component networks remain hidden from other networks. The **cost** *cost* keyword-argument pair specifies the metric or cost for the summary route, which is used during OSPF SPF calculations to determine the shortest paths to the destination. Valid values range from 0 to 16777215.
area *area-id* **nssa** **Example:** `ciscoasa(config-rtr)# area 1 nssa`	Specifies an NSSA area.

Command	Purpose
area *area-id* **stub** **Example:** `ciscoasa(config-rtr)# area 1 stub`	Specifies a stub area.
area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**ttl-security hops** *hop-count*] **Example:** `ciscoasa(config-rtr)# area 1 virtual-link 192.168.255.1 hello-interval 5`	Defines a virtual link and its parameters. The *area-id* argument identifies the area for which routes are to be summarized. The **virtual link** keyword specifies the creation of a virtual link neighbor. The *router-id* argument specifies the router ID that is associated with the virtual link neighbor. Enter the **show ospf** or **show ipv6 ospf** command to display the router ID. There is no default value. The **hello-interval** keyword specifies the time in seconds between the hello packets that are sent on an interface. The hello interval is an unsigned integer that is to be advertised in the hello packets. The value must be the same for all routers and access servers that are attached to a common network. Valid values range from 1 to 8192. The default is 10. The **retransmit-interval** *seconds* keyword-argument pair specifies the time in seconds between LSA retransmissions for adjacencies that belong to the interface. The retransmit interval is the expected round-trip delay between any two routers on the attached network. The value must be greater than the expected round-trip delay, and can range from 1 to 8192. The default is 5. The **transmit-delay** *seconds* keyword-argument pair specifies the estimated time in seconds that is required to send a link-state update packet on the interface. The integer value must be greater than zero. LSAs in the update packet have their own ages incremented by this amount before transmission. The range of values can be from 1 to 8192. The default is 1. The **dead-interval** *seconds* keyword-argument pair specifies the time in seconds that hello packets are not seen before a neighbor indicates that the router is down. The dead interval is an unsigned integer. The default is four times the hello interval, or 40 seconds. The value must be the same for all routers and access servers that are attached to a common network. Valid values range from 1 to 8192. The **ttl-security hops** keyword configures the time-to-live (TTL) security on a virtual link. The *hop-count* argument value can range from 1 to 254.

Configuring OSPFv3 Passive Interfaces

To configure OSPFv3 passive interfaces, perform the following steps:

	Command	Purpose
Step 1	**ipv6 router ospf** *process_id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**passive-interface** [*interface_name*] **Example:** `ciscoasa(config-rtr)# passive-interface inside`	Suppresses the sending and receiving of routing updates on an interface. The *interface_name* argument specifies the name of the interface on which the OSPFv3 process is running. If the *no interface_name* argument is specified, all of the interfaces in the OSPFv3 process *process_id* are made passive.

Configuring OSPFv3 Administrative Distance

To configure OSPFv3 administrative distance for IPv6 routes, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process_id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**distance** [**ospf** {**external** | **inter-area** | **intra-area**}] *distance* **Example:** `ciscoasa(config-rtr)# distance ospf external 200`	Sets the administrative distance for OSPFv3 routes. The **ospf** keyword specifies OSPFv3 routes. The **external** keyword specifies the external Type 5 and Type 7 routes for OSPFv3. The **inter-area** keyword specifies the inter-area routes for OSPVv3. The **intra-area** keyword specifies the intra-area routes for OSPFv3. The *distance* argument specifies the administrative distance, which is an integer from 10 to 254.

Configuring OSPFv3 Timers

You can set LSA arrival, LSA pacing, and throttling timers for OSPFv3.

To set the minimum interval at which the ASA accepts the same LSA from OSPFv3 neighbors, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**timers lsa arrival** *milliseconds* **Example:** `ciscoasa(config-rtr)# timers lsa arrival 2000`	Sets the minimum interval at which the ASA accepts the same LSA from OSPF neighbors. The *milliseconds* argument specifies the minimum delay in milliseconds that must pass between acceptance of the same LSA arriving from neighbors. The range is from 0 to 6,000,000 milliseconds. The default is 1000 milliseconds.

To configure LSA flood packet pacing, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**timers pacing flood** *milliseconds* **Example:** `ciscoasa(config-rtr)# timers lsa flood 20`	Configures LSA flood packet pacing. The *milliseconds* argument specifies the time in milliseconds at which LSAs in the flooding queue are paced in between updates. The configurable range is from 5 to 100 milliseconds. The default value is 33 milliseconds.

To change the interval at which OSPFv3 LSAs are collected into a group and refreshed, check summed, or aged, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**timers pacing lsa-group** *seconds* **Example:** `ciscoasa(config-rtr)# timers pacing lsa-group 300`	Changes the interval at which OSPFv3 LSAs are collected into a group and refreshed, checksummed, or aged. The *seconds* argument specifies the number of seconds in the interval at which LSAs are grouped, refreshed, check summed, or aged. The range is from 10 to 1800 seconds. The default value is 240 seconds.

To configure LSA retransmission packet pacing, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**timers pacing retransmission** *milliseconds* **Example:** `ciscoasa(config-rtr)# timers pacing retransmission 100`	Configures LSA retransmission packet pacing. The *milliseconds* argument specifies the time in milliseconds at which LSAs in the retransmission queue are paced. The configurable range is from 5 to 200 milliseconds. The default value is 66 milliseconds.

LSA and SPF throttling provide a dynamic mechanism to slow down LSA updates in OSPFv3 during times of network instability and allow faster OSPFv3 convergence by providing LSA rate limiting in milliseconds.

To configure LSA and SPF throttling timers, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	Choose one of the following options:	

Command	Purpose
timers throttle lsa *milliseconds1 milliseconds2 milliseconds3* **Example:** `ciscoasa(config-rtr)# timers throttle lsa 500 6000 8000`	Configures OSPFv3 LSA throttling. The *milliseconds1* argument specifies the delay in milliseconds to generate the first occurrence of the LSA. The *milliseconds2* argument specifies the maximum delay in milliseconds to originate the same LSA. The *milliseconds3* argument specifies the minimum delay in milliseconds to originate the same LSA. For LSA throttling, if the minimum or maximum time is less than the first occurrence value, then OSPFv3 automatically corrects to the first occurrence value. Similarly, if the maximum delay specified is less than the minimum delay, then OSPFv3 automatically corrects to the minimum delay value. The default values for LSA throttling are the following: • For *milliseconds1*, the default value is 0 milliseconds. • For *milliseconds2* and *milliseconds3*, the default value is 5000 milliseconds.
timers throttle spf *milliseconds1 milliseconds2 milliseconds3* **Example:** `ciscoasa(config-rtr)# timers throttle spf 5000 12000 16000`	Configures OSPFv3 SPF throttling. The *milliseconds1* argument specifies the delay in milliseconds to receive a change to the SPF calculation. The *milliseconds2* argument specifies the delay in milliseconds between the first and second SPF calculations. The *milliseconds3* argument specifies the maximum wait time in milliseconds for SPF calculations. For SPF throttling, if *milliseconds2* or *milliseconds3* is less than *milliseconds1*, then OSPFv3 automatically corrects to the *milliseconds1* value. Similarly, if *milliseconds3* is less than *milliseconds2*, then OSPFv3 automatically corrects to the *milliseconds2* value. The default values for SPF throttling are the following: • For *milliseconds1*, the default value is 5000 milliseconds. • For *milliseconds2* and *milliseconds3*, the default value is 10000 milliseconds.

Defining Static OSPFv3 Neighbors

You need to define static OSPFv3 neighbors to advertise OSPF routes over a point-to-point, non-broadcast network. This feature lets you broadcast OSPFv3 advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

Before you begin, you must create a static route to the OSPFv3 neighbor. See Chapter 27, "Static and Default Routes," for more information about creating static routes.

To define a static OSPFv3 neighbor, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**ipv6 ospf neighbor** *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**] **Example:** `ciscoasa(config-if)# interface ethernet0/0 ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01`	Configures OSPFv3 router interconnections to non-broadcast networks.

Resetting OSPFv3 Default Parameters

To return an OSPFv3 parameter to its default value, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**default** [**area** | **auto-cost** | **default-information** | **default-metric** | **discard-route** | **discard-route** | **distance** | **distribute-list** | **ignore** | **log-adjacency-changes** | **maximum-paths** | **passive-interface** | **redistribute** | **router-id** | **summary-prefix** | **timers**] **Example:** `ciscoasa(config-rtr)# default metric 5`	Returns an optional parameter to its default value. The **area** keyword specifies the OSPFv3 area parameters. The **auto-cost** keyword specifies the OSPFv3 interface cost according to bandwidth. The **default-information** keyword distributes default information. The **default-metric** keyword specifies the metric for a redistributed route. The **discard-route** keyword enables or disables the discard-route installation. The **distance** keyword specifies the administrative distance. The **distribute-list** keyword filters networks in routing updates. The **ignore** keyword ignores a specific event. The **log-adjacency-changes** keyword logs changes in the adjacency state. The **maximum-paths** keyword forwards packets over multiple paths. The **passive-interface** keyword suppresses routing updates on an interface. The **redistribute** keyword redistributes IPv6 prefixes from another routing protocol. The **router-id** keyword specifies the router ID for the specified routing process. The **summary-prefix** keyword specifies the IPv6 summary prefix. The **timers** keyword specifies the OSPFv3 timers.

Sending Syslog Messages

To configure the router to send a syslog message when an OSPFv3 neighbor goes up or down, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**log-adjacency-changes** [**detail**] **Example:** `ciscoasa(config-rtr)# log-adjacency-changes detail`	Configures the router to send a syslog message when an OSPFv3 neighbor goes up or down. The **detail** keyword sends a syslog message for each state, not only when an OSPFv3 neighbor goes up or down.

Suppressing Syslog Messages

To suppress the sending of syslog messages when the route receives unsupported LSA Type 6 multicast OSPF (MOSPF) packets, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `ciscoasa(config-if)# router ospf 1`	Enables an OSPFv2 routing process and enters router configuration mode. The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**ignore lsa mospf** **Example:** `ciscoasa(config-rtr)# ignore lsa mospf`	Suppresses the sending of syslog messages when the router receives unsupported LSA Type 6 MOSPF packets.

Calculating Summary Route Costs

To calculate summary route costs according to RFC 1583, enter the following command:

Command	Purpose
compatible rfc1583 **Example:** `ciscoasa (config-rtr)# compatible rfc1583`	Restores the methods that are used to calculate summary route costs according to RFC 1583.

Generating a Default External Route into an OSPFv3 Routing Domain

To generate a default route into an OSPFv3 routing domain, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**default-information originate** [**always**] **metric** *metric-value* [**metric-type** *type-value*] [**route-map** *map-name*] **Example:** `ciscoasa(config-rtr)# default-information originate always metric 3 metric-type 2`	Generates a default external route into an OSPFv3 routing domain. The **always** keyword advertises the default route whether or not the default route exists. The **metric** *metric-value* keyword-argument pair specifies the metric used for generating the default route. If you do not specify a value using the **default-metric** command, the default value is 10. Valid metric values range from 0 to 16777214. The **metric-type** *type-value* keyword-argument pair specifies the external link type that is associated with the default route that is advertised into the OSPFv3 routing domain. Valid values can be one of the following: • 1—Type 1 external route • 2—Type 2 external route The default is the type 2 external route. The **route-map** *map-name* keyword-argument pair specifies the routing process that generates the default route if the route map is satisfied.

Configuring an IPv6 Summary Prefix

To configure an IPv6 summary prefix, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process_id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.

Step 2	**summary-prefix** *prefix* [**not-advertise** | **tag** *tag-value*]
	
	Example:
	`ciscoasa(config-if)# ipv6 router ospf 1`
	`ciscoasa(config-rtr)# router-id 192.168.3.3`
	`ciscoasa(config-rtr)# summary-prefix FECO::/24`
	`ciscoasa(config-rtr)# redistribute static`
		Configures an IPv6 summary prefix.
		
		The *prefix* argument is the IPv6 route prefix for the destination. The **not-advertise** keyword suppresses routes that match the specified prefix and mask pair. This keyword applies to OSPFv3 only. The **tag** *tag-value* keyword-argument pair specifies the tag value that can be used as a match value for controlling redistribution through route maps. This keyword applies to OSPFv3 only.

Redistributing IPv6 Routes

To redistribute connected routes into an OSPFv3 process, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**ipv6 router ospf** *process-id* **Example:** `ciscoasa(config-if)# ipv6 router ospf 1`	Enables an OSPFv3 routing process and enters IPv6 router configuration mode. The *process-id* argument is an internally used identifier for this routing process, is locally assigned, and can be any positive integer from 1 to 65535. This ID does not have to match the ID on any other device; it is for internal administrative use only. You can use a maximum of two processes.
Step 2	**redistribute** *source-protocol* [*process-id*] [**include-connected** {[**level-1** | **level-2**}] [*as-number*] [**metric** [*metric-value* | **transparent**}] **[metric-type** *type-value*] [**match** {**external** [**1**|**2**] | **internal** | **nssa-external** [**1**|**2**]}] [**tag** *tag-value*] [**route-map** *map-tag*] **Example:** `ciscoasa(config-rtr)# redistribute connected 5 type-1`	Redistributes IPv6 routes from one OSPFv3 process into another. The *source-protocol* argument specifies the source protocol from which routes are being redistributed, which can be static, connected, or OSPFv3. The *process-id* argument is the number that is assigned administratively when the OSPFv3 routing process is enabled. The **include-connected** keyword allows the target protocol to redistribute routes learned by the source protocol and connected prefixes on those interfaces over which the source protocol is running. The **level-1** keyword specifies that for Intermediate System-to-Intermediate System (IS-IS), Level 1 routes are redistributed into other IP routing protocols independently. The **level-1-2** keyword specifies that for IS-IS, both Level 1 and Level 2 routes are redistributed into other IP routing protocols. The **level-2** keyword specifies that for IS-IS, Level 2 routes are redistributed into other IP routing protocols independently. For the **metric** *metric-value* keyword-argument pair, when redistributing routes from one OSPFv3 process into another OSPFv3 process on the same router, the metric is carried through from one process to the other if no metric value is specified. When redistributing other processes into an OSPFv3 process, the default metric is 20 when no metric value is specified. The **metric transparent** keyword causes RIP to use the routing table metric for redistributed routes as the RIP metric. The **metric-type** *type-value* keyword-argument pair specifies the external link type that is associated with the default route that is advertised into the OSPFv3 routing domain. Valid values can be one of the following: 1 for a Type 1 external route or 2 for a Type 2 external route. If no value is specified for the **metric-type** keyword, the ASA adopts a Type 2 external route. For IS-IS, the link type can be one of the following: internal for an IS-IS metric that is less than 63 or external for an IS-IS metric that is greater than 64 and less than 128. The default is internal. The **match** keyword redistributes routes into other routing domains and is used with one of the following options: **external** [**1**|**2**] for routes that are external to the autonomous system, but are imported into OSPFv3 as Type 1 or Type 2 external routes; **internal** for routes that are internal to a specific autonomous system; **nssa-external** [**1**|**2**] for routes that are external to the autonomous system, but are imported into OSPFv3 in an NSSA for IPv6 as Type 1 or Type 2 external routes. The **tag** *tag-value* keyword-argument pair specifies the 32-bit decimal value that is attached to each external route, which may be used to communicate information between ASBRs. If none is specified, then the remote autonomous system number is used for routes from BGP and EGP. For other protocols, zero is used. Valid values range from 0 to 4294967295. The **route-map** keyword specifies the route map to check for filtering the importing of routes from the source routing protocol to the current routing protocol. If this keyword is not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes are imported. The *map-tag* argument identifies a configured route map.

Removing the OSPF Configuration

To remove the entire OSPFv2 configuration that you have already enabled, enter the following command:

Command	Purpose
clear configure router ospf *pid* **Example:** `ciscoasa(config)# clear configure router ospf 1000`	Removes the entire OSPFv2 configuration that you have enabled. After the configuration is cleared, you must reconfigure OSPF using the **router ospf** command.

To remove the entire OSPFv3 configuration that you have already enabled, enter the following command:

Command	Purpose
clear configure ipv6 router ospf *process-id* **Example:** `ciscoasa(config)# clear configure ipv6 router ospf 1000`	Removes the entire OSPFv3 configuration that you have enabled. After the configuration is cleared, you must reconfigure OSPFv3 using the **ipv6 router ospf** command.

Configuration Example for OSPFv2

The following example shows how to enable and configure OSPFv2 with various optional processes:

Step 1 To enable OSPFv2, enter the following commands:

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

Step 2 (Optional) To redistribute routes from one OSPFv2 process to another OSPFv2 process, enter the following commands:

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

Step 3 (Optional) To configure OSPFv2 interface parameters, enter the following commands:

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
ciscoasa(config-rtr)# interface inside
ciscoasa(config-interface)# ospf cost 20
ciscoasa(config-interface)# ospf retransmit-interval 15
ciscoasa(config-interface)# ospf transmit-delay 10
ciscoasa(config-interface)# ospf priority 20
ciscoasa(config-interface)# ospf hello-interval 10
```

```
ciscoasa(config-interface)# ospf dead-interval 40
ciscoasa(config-interface)# ospf authentication-key cisco
ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco
ciscoasa(config-interface)# ospf authentication message-digest
```

Step 4 (Optional) To configure OSPFv2 area parameters, enter the following commands:

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# area 0 authentication
ciscoasa(config-rtr)# area 0 authentication message-digest
ciscoasa(config-rtr)# area 17 stub
ciscoasa(config-rtr)# area 17 default-cost 20
```

Step 5 (Optional) To configure the route calculation timers and show the log neighbor up and down messages, enter the following commands:

```
ciscoasa(config-rtr)# timers spf 10 120
ciscoasa(config-rtr)# log-adj-changes [detail]
```

Step 6 (Optional) To show current OSPFv2 configuration settings, enter the **show ospf** command.

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

```
ciscoasa(config)# show ospf

Routing Process "ospf 2" with ID 10.1.89.2 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
```

Step 7 To clear the OSPFv2 configuration, enter the following command:

```
ciscoasa(config)# clear configure router ospf pid
```

Configuration Examples for OSPFv3

The following example shows how to enable and configure OSPFv3 at the interface level:

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf 1 area 1
```

The following is sample output from the **show running-config ipv6** command:

```
ciscoasa (config)# show running-config ipv6
ipv6 router ospf 1
 log-adjacency-changes
```

The following is sample output from the **show running-config interface** command:

```
ciscoasa (config-if)# show running-config interface GigabitEthernet3/1
interface GigabitEthernet3/1
 nameif fda
 security-level 100
 ip address 1.1.11.1 255.255.255.0 standby 1.1.11.2
 ipv6 address 9098::10/64 standby 9098::11
 ipv6 enable
 ipv6 ospf 1 area 1
```

The following examples show how to configure OSPFv3-specific interfaces:

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# nameif fda
ciscoasa (config-if)# security-level 100
ciscoasa (config-if)# ip address 10.1.11.1 255.255.255.0 standby 10.1.11.2
ciscoasa (config-if)# ipv6 address 9098::10/64 standby 9098::11
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf cost 900
ciscoasa (config-if)# ipv6 ospf hello-interval 20
ciscoasa (config-if)# ipv6 ospf network broadcast
ciscoasa (config-if)# ipv6 ospf database-filter all out
ciscoasa (config-if)# ipv6 ospf flood-reduction
ciscoasa (config-if)# ipv6 ospf mtu-ignore
ciscoasa (config-if)# ipv6 ospf 1 area 1 instance 100
ciscoasa (config-if)# ipv6 ospf encryption ipsec spi 890 esp null md5
12345678901234567890123456789012

ciscoasa (config)# ipv6 router ospf 1
ciscoasa (config)# area 1 nssa
ciscoasa (config)# distance ospf intra-area 190 inter-area 100 external 100
ciscoasa (config)# timers lsa arrival 900
ciscoasa (config)# timers pacing flood 100
ciscoasa (config)# timers throttle lsa 900 900 900
ciscoasa (config)# passive-interface fda
ciscoasa (config)# log-adjacency-changes
ciscoasa (config)# redistribute connected metric 100 metric-type 1 tag 700
```

For an example of how to configure an OSPFv3 virtual link, see the following URL:

http://www.cisco.com/en/US/tech/tk365/technologies_configuration_example09186a0080b8fd06.shtml

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can also 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 monitor or display various OSPFv2 routing statistics, enter one of the following commands:

Command	Purpose
show ospf [*process-id* [*area-id*]]	Displays general information about OSPFv2 routing processes.
show ospf border-routers	Displays the internal OSPFv2 routing table entries to the ABR and ASBR.
show ospf [*process-id* [*area-id*]] **database**	Displays lists of information related to the OSPFv2 database for a specific router.
show ospf flood-list *if-name*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPF v2packet pacing). OSPFv2 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 of an interface. Pacing enables OSPFv2 update and retransmission packets to be sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.
show ospf interface [*if_name*]	Displays OSPFv2-related interface information.
show ospf neighbor [*interface-name*] [*neighbor-id*] [**detail**]	Displays OSPFv2 neighbor information on a per-interface basis.
show ospf request-list *neighbor if_name*	Displays a list of all LSAs requested by a router.
show ospf retransmission-list *neighbor if_name*	Displays a list of all LSAs waiting to be resent.
show ospf [*process-id*] **summary-address**	Displays a list of all summary address redistribution information configured under an OSPFv2 process.

Command	Purpose
show ospf [*process-id*] **traffic**	Displays a list of different types of packets being sent or received by a specific OSPFv2 instance.
show ospf [*process-id*] **virtual-links**	Displays OSPFv2-related virtual links information.
show route cluster	Displays additional OSPFv2 route synchronization information in clustering.

To monitor or display various OSPFv3 routing statistics, enter one of the following commands:

Command	Purpose
show ipv6 ospf [*process-id* [*area-id*]]	Displays general information about OSPFv3 routing processes.
show ipv6 ospf [*process-id*] **border-routers**	Displays the internal OSPFv3 routing table entries to the ABR and ASBR.
show ipv6 ospf [*process-id* [*area-id*]] **database** [**external** | **inter-area prefix** | **inter-area-router** | **network** | **nssa-external** | **router** | **area** | **as** | **ref-lsa** | [*destination-router-id*] [**prefix** *ipv6-prefix*] [*link-state-id*]] [**link** [**interface** *interface-name*] [**adv-router** *router-id*] | **self-originate**] [**internal**] [**database-summary**]	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospf [*process-id* [*area-id*]] **events**	Displays OSPFv3 event information.
show ipv6 ospf [*process-id*] [*area-id*] **flood-list** *interface-type interface-number*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPFv3 packet pacing). OSPFv3 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 retransmissions to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out of an interface. Pacing enables OSPFv3 update and retransmission packets to be sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.

Command	Purpose
show ipv6 ospf [*process-id*] [*area-id*] **interface** [*type number*] [**brief**]	Displays OSPFv3-related interface information.
show ipv6 ospf neighbor [*process-id*] [*area-id*] [*interface-type interface-number*] [*neighbor-id*] [**detail**]	Displays OSPFv3 neighbor information on a per-interface basis.
show ipv6 ospf [*process-id*] [*area-id*] **request-list** [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs requested by a router.
show ipv6 ospf [*process-id*] [*area-id*] **retransmission-list** [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs waiting to be resent.
show ipv6 ospf statistic [*process-id*] [**detail**]	Displays various OSPFv3 statistics.
show ipv6 ospf [*process-id*] **summary-prefix**	Displays a list of all summary address redistribution information configured under an OSPFv3 process.
show ipv6 ospf [*process-id*] **timers** [**lsa-group** | **rate-limit**]	Displays OSPFv3 timers information.
show ipv6 ospf [*process-id*] **traffic** [*interface_name*]	Displays OSPFv3 traffic-related statistics.
show ipv6 ospf virtual-links	Displays OSPFv3-related virtual links information.
show ipv6 route cluster [**failover**] [**cluster**] [**interface**] [**ospf**] [**summary**]	Displays the IPv6 routing table sequence number, IPv6 reconvergence timer status, and IPv6 routing entries sequence number in a cluster.

Additional References

For additional information related to implementing OSPF, see the following section:

- RFCs

RFCs

RFC	Title
2328	OSPFv2
4552	OSPFv3 Authentication
5340	OSPF for IPv6

Feature History for OSPF

Table 30-1 lists each feature change and the platform release in which it was implemented.

Table 30-1 Feature History for OSPF

Feature Name	Platform Releases	Feature Information
OSPF Support	7.0(1)	Support was added for route data, authentication, and redistribution and monitoring of routing information using the Open Shortest Path First (OSPF) routing protocol. We introduced the following command: **route ospf**
Dynamic Routing in Multiple Context Mode	9.0(1)	OSPFv2 routing is supported in multiple context mode.
Clustering		For OSPFv2 and OSPFv3, bulk synchronization, route synchronization, and Spanned EtherChannel load balancing are supported in the clustering environment. We introduced or modified the following commands: **show route cluster**, **show ipv6 route cluster**, **debug route cluster, router-id cluster-pool**.
OSPFv3 Support for IPv6		OSPFv3 routing is supported for IPv6. We introduced or modified the following commands: **ipv6 ospf**, **ipv6 ospf area**, **ipv6 ospf cost, ipv6 ospf database-filter all out, ipv6 ospf dead-interval, ipv6 ospf encryption, ipv6 ospf hello-interval, ipv6 ospf mtu-ignore, ipv6 ospf neighbor, ipv6 ospf network, ipv6 ospf flood-reduction, ipv6 ospf priority, ipv6 ospf retransmit-interval, ipv6 ospf transmit-delay, ipv6 router ospf, ipv6 router ospf area, ipv6 router ospf default, ipv6 router ospf default-information, ipv6 router ospf distance, ipv6 router ospf exit, ipv6 router ospf ignore, ipv6 router ospf log-adjacency-changes, ipv6 router ospf no, ipv6 router ospf passive-interface, ipv6 router ospf redistribute, ipv6 router ospf router-id, ipv6 router ospf summary-prefix, ipv6 router ospf timers, area encryption, area range, area stub, area nssa, area virtual-link, default, default-information originate, distance, ignore lsa mospf, log-adjacency-changes, redistribute, router-id, summary-prefix, timers lsa arrival, timers pacing flood, timers pacing lsa-group, timers pacing retransmission, timers throttle, show ipv6 ospf, show ipv6 ospf border-routers, show ipv6 ospf database, show ipv6 ospf events, show ipv6 ospf flood-list, show ipv6 ospf graceful-restart, show ipv6 ospf interface, show ipv6 ospf neighbor, show ipv6 ospf request-list, show ipv6 ospf retransmission-list, show ipv6 ospf statistic, show ipv6 ospf summary-prefix, show ipv6 ospf timers, show ipv6 ospf traffic, show ipv6 ospf virtual-links, show ospf, show running-config ipv6 router, clear ipv6 ospf, clear configure ipv6 router, debug ospfv3, ipv6 ospf neighbor**.

Table 30-1 Feature History for OSPF (continued)

Feature Name	Platform Releases	Feature Information
OSPF support for Fast Hellos	9.2(1)	OSPF supports the Fast Hello Packets feature, resulting in a configuration that results in faster convergence in an OSPF network. We modified the following command: **ospf dead-interval**
Timers		New OSPF timers were added; old ones were deprecated. We introduced the following commands: **timers lsa arrival**, **timers pacing**, **timers throttle** We removed the following commands: **Timers spf**, **timers lsa-grouping-pacing**
Route filtering using access-list		Route filtering using ACL is now supported. We introduced the following command: **distribute-list** We introduced the following screen:
OSPF Monitoring enhancements		Additional OSPF monitoring information was added. We modified the following commands: **show ospf events**, **show ospf rib**, **show ospf statistics**, **show ospf border-routers [detail]**, **show ospf interface brief**
OSPF redistribute BGP		OSPF redistribution feature was added. We added the following command: **redistribute bgp**