Service Advertisement Framework Configuration Guide, Cisco IOS Release 15SY

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring Cisco SAF

Cisco Service Advertisement Framework (SAF) provides a framework that allows applications to discover the existence, location, and configuration of networked resources within networks. Cisco SAF allows a timely and reliable awareness of the services within networks, while applications advertise and discover services on networks. Service information distributes through a network of Cisco SAF cooperative nodes that assume specific functions to efficiently distribute knowledge of services and facilitate their discovery.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Cisco SAF

- Before configuring Cisco SAF, you should understand the concepts in this guide.
- Before configuring neighbor relationships for Cisco SAF Forwarders located on separate LANs, ensure that IP routing is configured between each Cisco SAF Forwarder.

Restriction for Cisco SAF

- Cisco SAF works independently of Cisco EIGRP routing.

Information About Cisco SAF

Cisco SAF Overview

Cisco Service Advertisement Framework (SAF) provides a framework that allows applications to discover the existence, location, and configuration of networked resources within networks. Cisco SAF allows a timely and reliable awareness of the services within networks, while applications advertise and discover services on networks. Service information distributes through a network of Cisco SAF cooperative nodes that assume specific functions to efficiently distribute knowledge of services and facilitate their discovery.

A non-SAF node is any node in a network that does not understand SAF. Non-SAF nodes are called “dark nets” and are required to traverse ISPs. Cisco SAF messages are IP-based and therefore are unaffected by dark nets.

Cisco SAF cooperative network nodes are grouped into two major functional responsibilities:

- Cisco SAF forwarder—Distributes service information through the network and makes these services discoverable by clients in the network.
- Cisco SAF client—Advertises and discovers services.

An effective Cisco SAF network requires both roles to be configured.

Cisco SAF Forwarder Overview

A Cisco SAF forwarder receives services advertised by Cisco SAF Clients, distributes the services reliably through the network, and makes services available for Cisco SAF clients. A Cisco SAF forwarder:

- Ensures reliable delivery of service advertisements.
- Maintains knowledge of path redundancy.
- Is scalable.
- Minimizes the use of network bandwidth by using targeted multicast and unicast messages.

The Cisco SAF forwarder can propagate service advertisements to other Cisco SAF forwarders and can propagate across a LAN, campus network, WAN, or ISP.

A basic Cisco SAF forwarder provides the relationship between Cisco SAF clients and the framework. A Cisco SAF forwarder is normally located at the edges or boundaries of a network. The Cisco SAF forwarder receives service advertisements and stores a copy before forwarding the advertisement to its neighbor SAF nodes. The client and forwarder relationship is to maintain the advertisement. If a client removes a service or disconnects from the forwarder node, the node will inform the framework about the services that are no longer available. When the forwarder node receives advertisements from other

forwarder nodes, it will keep a copy of the entire advertisement (header and opaque data) and forward it to other SAF peers.

You can configure a Cisco SAF forwarder on a LAN to automatically allow dynamic discovery of services to all enabled interfaces and, at the same time, specify interfaces (static configuration) that you want blocked to other interfaces attempting to discover their services.

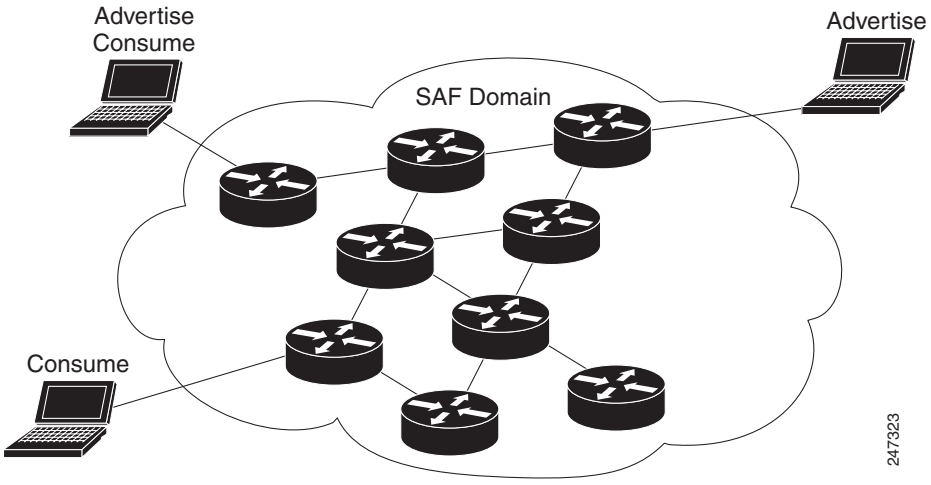
You can configure a Cisco SAF forwarder across a non-SAF node to automatically allow dynamic discovery of services. For example, Cisco SAF forwarders can join a common peer group. You can also create static configurations (unicast) between pairs of Cisco SAF forwarders.

Note Multicast routing is required to allow dynamic discovery of services.

Note With CSCts55778 and CSCtq71851, the Cisco SAF forwarder can send service metadata to its neighbor SAF nodes. Metadata is XML information, and service data is information that a server communicates to a client about itself. The service metadata does not propagate in mixed software release environments until such time that the version of EIGRP and SAF is upgraded.

Cisco SAF Client Overview

A Cisco SAF client is a producer (advertises to the network) or a consumer of services (requests a service from the network), or both. When a Cisco SAF client sends a register message to a Cisco SAF forwarder, it establishes a relationship with the Cisco SAF forwarder. The Cisco SAF forwarder uses this register message to obtain a unique handle that distinctly identifies this Cisco SAF client from others connected to the same forwarder. Only after a Cisco SAF client registers is it able to advertise (publish) or request (subscribe to) services. The figure below shows a typical Cisco SAF network.



When advertising a service, a Cisco SAF client publishes (sends) advertisements, to the Cisco SAF forwarder, that contain information about the service it offers. Services are identified by a unique service ID, sub-service ID, and instance ID and are described by service data. For more information on service identifiers, see Cisco SAF Service Identifier Number Formats, page 5. The Cisco SAF client can send multiple publish requests, each advertising a distinct service. The Cisco SAF forwarder advertises all services published by the Cisco SAF client. The Cisco SAF client can update an existing service

advertisement by sending a new publish request for the same service. The client can also generate an unpublish request, which removes one of its existing service advertisements.

When requesting a service, the Cisco SAF client sends a request notification of services using a subscribe request. The subscribe request contains a filter that describes the set of services in which the Cisco SAF client is interested. In response to this request, the Cisco SAF forwarder sends the current set of services that match the filter to the Cisco SAF client in a series of notify requests. As with a publish request, the Cisco SAF client can generate multiple subscribe requests, each with a different filter. The Cisco SAF client can also generate an unsubscribe request, which removes one of its existing subscriptions.

Cisco SAF clients can be internal (existing within a Cisco SAF forwarder) or external (existing on a separate device and communicating with a Cisco SAF forwarder using the XMCP protocol). Internal Cisco SAF clients include Capabilities Manager (see How to Configure Capabilities Manager, page 60) and Cisco Unified Communications Manager Express (Cisco Unified CME). External Cisco SAF clients using XMCP include Cisco Unified Communications Manager.

External Cisco SAF Client Using XMCP Overview

An external Cisco SAF client initiates a TCP connection to a Cisco SAF forwarder that has been configured as an XMCP server. Once the TCP connection is established, the client begins an XMCP session over this connection by sending an XMCP register message to the Cisco SAF forwarder.

An XMCP session uses a username and password for security.

- The username is included in requests from the XMCP client (Cisco SAF client) to the XMCP server (Cisco SAF forwarder).
- The password is a shared secret that is not sent in requests, but is used by the client to compute a message-integrity value that is appended to the request.

When an XMCP server receives a request, it locates the username attribute in that request and uses it to access its local copy of the password, and then it computes its own message-integrity value for the request. If the computations match, the passwords must match and the request is authenticated, permitting the XMCP client to act as a Cisco SAF client. If they do not match, the password is incorrect and the request will be rejected.

Once the XMCP session has been established successfully, the XMCP client may send XMCP publish, unpublish, subscribe, and unsubscribe requests. When the server receives and successfully authenticates these requests, it translates the requests into the equivalent Cisco SAF client requests and sends them to the Cisco SAF forwarder. Similarly, Cisco SAF client notify requests from the forwarder are translated into XMCP notify requests and sent to the XMCP client.

Because an external Cisco SAF client may lose connectivity to the Cisco SAF network in the event of a network outage, a Cisco SAF forwarder requires periodic verification regarding the liveliness of the Cisco SAF client to advertise its services to the Cisco SAF network. In XMCP, this is accomplished by exchanging a liveliness timer between the client and the server at the time of registration. The XMCP client is responsible for ensuring that the interval between requests never exceeds this value. An XMCP client has no data (publish or subscribe) to send, so it generates a small keepalive message to refresh the timer on the server.

A Cisco SAF forwarder considers that an external Cisco SAF client has failed if it has not seen an XMCP request from the client in a time period equal to the liveliness timer. When a Cisco SAF forwarder detects that the Cisco SAF client has failed, it withdraws the services advertised on behalf of that Cisco SAF client from the network and removes any subscriptions that the Cisco SAF client had established. As an alternative to waiting for the liveliness timer to expire, a Cisco SAF client can be manually unregistered (sending an unregister request to terminate the XMCP session) to gracefully cause a Cisco SAF forwarder to withdraw all services and subscriptions.

Cisco SAF Service Identifier Number Formats

A service is any information that a Cisco SAF client application wishes to advertise, which can then be used by other Cisco SAF client applications. A service advertisement consists of service data. Service advertisements are propagated between forwarders using header data. Cisco SAF clients that are interested in a service receive, and may inspect, service header and service data.

A service identifier number uniquely identifies the service on a network. The following example shows the format of a service identifier number:

```
service:sub-service:instance.instance.instance.instance
```

The service identifier is a 16-bit decimal identifier for the major service being advertised. A major service refers to a specific technology area, such as Cisco Unified Communications (UC). Service identifiers are assigned by Cisco to various customers who require an SAF client.

The following example shows the service ID values for IP Everywhere and Cisco Unified Communications:

```
Cisco Defined Numbers
    SAF_SERVICE_ID_IPE                = 100      ! IP Everywhere
    SAF_SERVICE_ID_UC                 = 101      ! Unified Communications
```

The sub-service identifier is a 16-bit decimal identifier for the minor service being advertised. A sub-service (also referred to as a minor service) refers to the type of service within a technology. For example, within UC:

- Sub-service 1 is TDM gateway.
- Sub-service 2 is hosted-DN.
- Instance identifies a specific service advertisement for this kind of service.

For example, service identifier 101:1:abcd.1234.ef.678 could be an advertisement of a UC (service 101) TDM gateway (sub-service 1) announced by the Communications Manager cluster in a certain location (instance abcd.1234.ef.678). The instance identifier is a unique 128-bit number that identifies the specific service advertised.

Client teams define the use of sub-service and instance values for their applications. Clients must ensure instance uniqueness within a Cisco SAF domain.

Cisco SAF and the Role of Domains in a Network

As the variety and number of network services grow, providing timely and reliable awareness of these services starts to play a more significant role in increasing productivity and efficiency. One of the biggest challenges in propagating service availability awareness over a WAN is one of scalability. As networks grow, the services offered by the devices on these networks increase. Protocols responsible for the service advertisement need to scale to handle this increased load. These protocols also need to react efficiently to rapid changes and propagate the new information in a timely manner.

Cisco SAF is designed to be a scalable solution for enterprise service locations and is capable of spanning LAN and WAN Internet segments. As an enterprise solution, you can configure Cisco SAF to use domains to scale for very large networks. Just as Cisco Enhanced Interior Gateway Routing Protocol (EIGRP) defines the concept of an autonomous system in which routes can be searched for in a hierarchical manner, Cisco SAF employs the similar concept of a domain and subdomains.

Cisco SAF provides a dynamic peer discovery and service advertisement propagation technique known as IP multicast. IP multicast requires the cooperation of IP Cisco SAF forwarders (the devices that connect IP subnets together to form intranets). IP multicasting, however, may not be completely implemented across

some intranets. In the absence of IP multicasting, Cisco SAF operates within the configured subnet or within the groups of subnets over which IP multicast is supported.

Cisco SAF forwarders offer two primary types of administrative domains (ADs); a domain and a subdomain. A domain and a subdomain function the same with one notable exception; subdomains do not form unique neighbor relationships, but instead rely on a single peering.

Ideally, a network would require only a single domain for advertising all services. However, because of scaling and policy issues, some networks require the creation of multiple domains. The recommendation is to use a single domain. Consider using multiple domains when:

- More than 30,000 services are registered in a single domain.
- Logical grouping of services is needed to restrict propagation of services.

Closed groups are needed to prevent users from browsing services that they are not allowed to access.

Service redistribution allows different domains to exchange service information. Services may need to be bound to specific areas of the network, or the number of services in a given network may need to be limited. If you cannot use a single domain, service advertisement redistribution might be the solution.

Each domain on a network is separated into an AD. All Cisco SAF forwarders in the same AD (running the same domain) have complete knowledge of the entire AS. A Cisco forwarder that connects two (or more) administrative domains is known as a border forwarder. A border forwarder advertises service information from one AS to another AS. Proper design should also be considered if multiple border forwarders are used to avoid loops (information learned from one AD being sent back to the same AD).

Cisco SAF Virtual Routers

Cisco EIGRP service-family support extends the named configuration to allow configuration of multiple instances that operate independently. The addition of a Virtual Router ID (VRID) to the base Cisco EIGRP packet encoding allows for multiple instances.

As each virtual device is created, a VRID is assigned to the top-level router and shared with the address families and service families that are configured under it.

Cisco SAF Neighbor Relationships

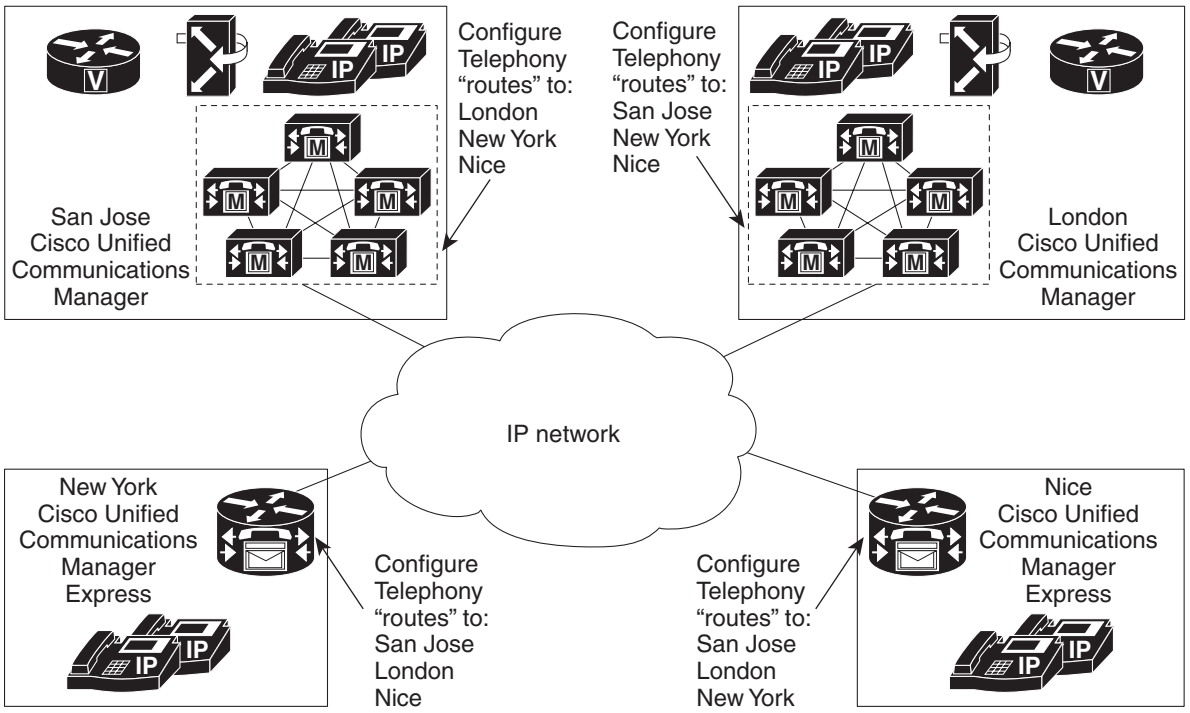
Cisco SAF forwarders can operate in networks that do not have devices that support the Cisco SAF forwarder protocol. These networks are referred to as “dark nets.” There are two methods for configuring Cisco SAF forwarders over IP networks that do not support Cisco SAF (IP clouds); unicast Cisco SAF neighbors and multicast Cisco SAF neighbors.

You can use a unicast configuration to provide a reliable point-to-point adjacency with neighbors. As the number of Cisco SAF forwarders increases, you can use multicast to provide an efficient transport between multiple Cisco SAF neighbors. A single IP multicast group address can be used for multiple Cisco SAF neighbors to exchange SAF information in a peer-group.

Benefits of Cisco SAF

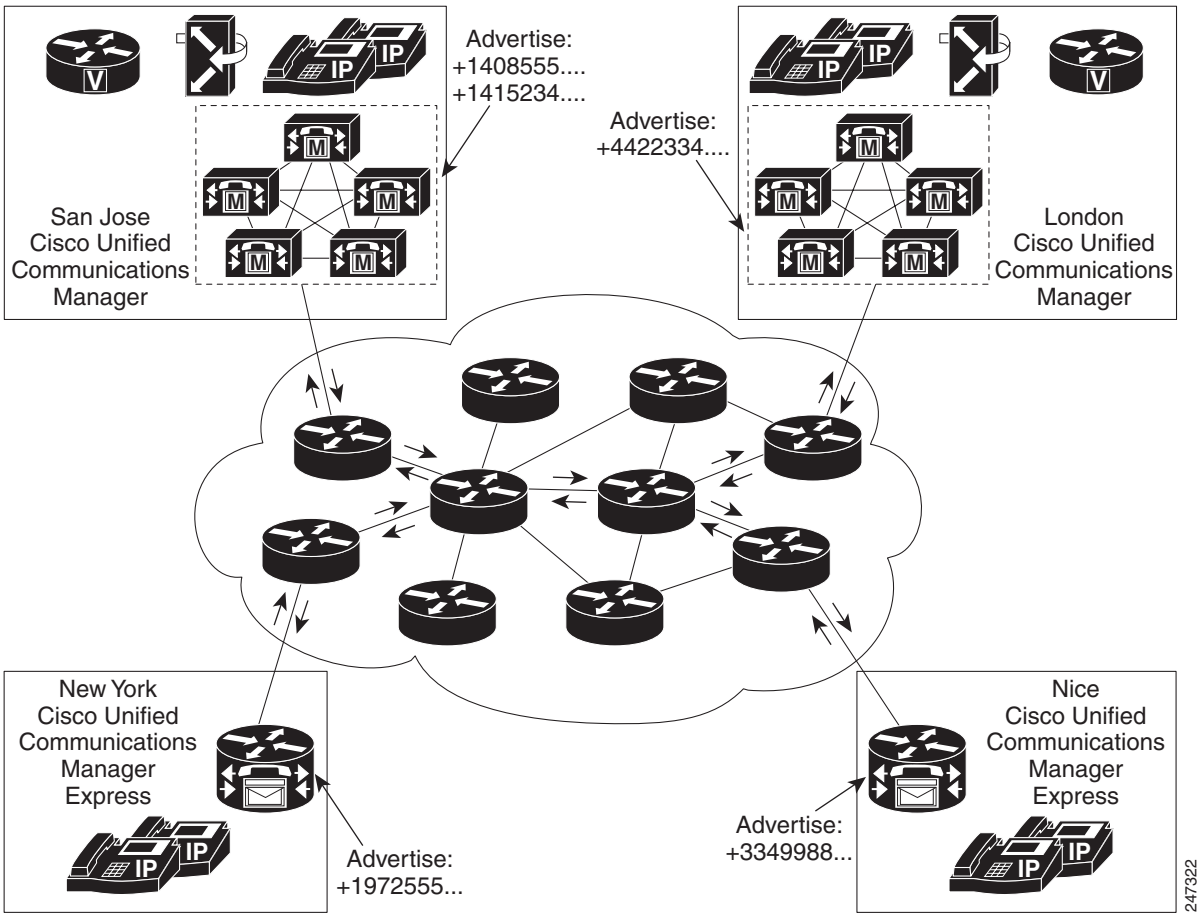
Traditionally, to locate services on a network, network applications must be configured with the hostname and the network address of the desired service or must use an overlay mechanism such as DNS. Existing protocols that support service advertisement provide periodic-based announcements of resource utilization. These network services are typically LAN-based.

The figure below shows a Cisco Unified Communications Manager network that requires a traditional configuration methodology.



Cisco SAF provides a framework that allows networking applications to automatically discover the existence, location, and configuration of networked services within networks. This automated discovery of services replaces the manual entry of complex configurations such as dial plans that often require repetitive configuration changes. Cisco SAF also allows applications to advertise and discover their services. Cisco SAF allows you to create a configuration once and then have it propagate to all devices that require the information.

The figure below shows a Cisco Unified Communications Manager network that uses Cisco SAF.



You can configure a Cisco SAF client either on the same device as the Cisco SAF forwarder or on an external device.

How to Configure a Cisco SAF Forwarder

- Displaying Cisco SAF Statistics, page 33
- Deleting Information from a Cisco SAF Configuration, page 38

Enabling Cisco SAF

To enable Cisco SAF and create a Cisco SAF service-discovery process, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **exit-service-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4**	**ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables Cisco SAF service family for the specified autonomous system.

	Command or Action	Purpose
Step 5	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

Configuring Interface-Specific Commands for Cisco SAF

Cisco SAF provides an inheritance precedence for interface-specific commands. Configurations made in service-family interface configuration mode have priority over specific service-family interface and factory default configurations. To configure interface-specific commands under the service family for Cisco SAF, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **sf-interface**
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.

	Command or Action	Purpose
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode snd creates a Cisco SAF service family for the specified autonomous system on the device, which is enabled by default.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet 0/0`	Enters service-family configuration mode and enables service-family interface configuration mode for the specified interface.
Step 6	**sf-interface** **Example:** `Device(config-router-sf-interface)# sf-interface hello-interval 10`	Enter the appropriate **interface** commands required for your configuration.
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Configuring Cisco SAF for Multi-Topology Networks

Use the following configuration to register clients and publish or subscribe services into a named topology. If you configure a second topology using an existing topology name, but with a different ID, it will replace the existing topology, rather than create two IDs for the same topology.

Note With CSCts55778 and CSCtq71851, the Cisco SAF forwarder can send service metadata to its neighbor SAF nodes. Metadata is XML information, and service data is information that a server communicates to a client about itself. The service metadata does not propagate in mixed software release environments until such time that the version of EIGRP and SAF is upgraded.

To configure Cisco SAF for multi-topology networks, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology base**
6. **exit-sf-topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**topology base** **Example:** `Device(config-router-sf)# topology base`	Enables service-family interface topology configuration mode and creates a topology base for the specified service-family interface.

	Command or Action	Purpose
Step 6	**exit-sf-topology** **Example:** `Device(config-router-sf-topology)# exit-sf-topology`	Exits service-family interface topology configuration mode.

Configuring Static Neighbor Relationships for Cisco SAF

Use the following commands to configure static neighbor adjacencies between Cisco SAF forwarders.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **neighbor** {*ip-address interface-type interface-number* | **description** *word* | **maximum-service** | **maximum-service** *number threshold-value* [**dampened** | **reset-time** | **restart** *interval* | **restart-count** | **warning-only**]}
6. **exit-service-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance in global configuration mode.

	Command or Action	Purpose
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**neighbor** {*ip-address interface-type interface-number* | **description** *word* | **maximum-service** | **maximum-service** *number threshold-value* [**dampened** | **reset-time** | **restart** *interval* | **restart-count** | **warning-only**]} **Example:** `Device(config-router-sf)# neighbor 10.10.10.1 Ethernet 0/0`	Enables a Cisco SAF neighbor relationship for the specified interface.
Step 6	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

Configuring Stub Routing for Cisco SAF

You can configure a Cisco SAF forwarder as a stub device. For complete information on Cisco EIGRP stub routing, see the “Configuring EIGRP” module in the *IP Routing: EIGRP Configuration Guide*.

To create an Cisco SAF stub device, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *number*
5. **eigrp stub** [**receive-only | connected**]
6. **exit-service-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instancee.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**eigrp stub** [**receive-only** | **connected**] **Example:** `Device(config-router-sf)# eigrp stub connected`	Configures a stub device for Cisco SAF.
Step 6	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

Configuring Route Authentication for Cisco SAF

Cisco SAF route authentication provides Message Digest 5 (MD5) authentication of routing updates from the routing protocol. The MD5 keyed digest in each packet prevents the introduction of unauthorized or false routing messages from unapproved sources. To configure route authentication for Cisco SAF, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **authentication key-chain** *name-of-chain*
7. **authentication mode** {**hmac-sha-256** {**0** | **7**} *password* | **md5**}
8. **exit-sf-interface**
9. **exit-service-family**
10. **exit**
11. **key-chain** *name-of-chain*
12. **key** *key-id*
13. **key-string** *text*
14. **accept-lifetime** *start-time* [**local** {**duration** *seconds* | **end-time** | **infinite**}]
15. **send-lifetime** *start-time* [**local** {**duration** *seconds* | **end-time** | **infinite**}]
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.

	Command or Action	Purpose
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet 0/0`	Enters service-family configuration mode and enables IPv4 service-family interface configuration mode for the specified interface.
Step 6	**authentication key-chain** *name-of-chain* **Example:** `Device(config-router-sf-interface)# authentication key-chain example`	Specifies an authentication key chain for EIGRP.
Step 7	**authentication mode** {**hmac-sha-256** {**0** | **7**} *password* | **md5**} **Example:** `Device(config-router-sf-interface)# authentication mode md5`	Enables IPv4 service-family authentication mode HMAC-SHA-256 or MD5 for the specified interface.
Step 8	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.
Step 9	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.

	Command or Action	Purpose
Step 11	**key-chain** *name-of-chain* **Example:** `Device(config)# key-chain example`	Defines an authentication key chain needed to enable authentication for routing protocols and enters key-chain configuration mode.
Step 12	**key** *key-id* **Example:** `Device(config-keychain)# key example`	Enters key configuration mode and identifies an authentication string for a key.
Step 13	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string example`	Specifies the authentication string for a key.
Step 14	**accept-lifetime** *start-time* [**local** {**duration** *seconds* | **end-time** | **infinite**}] **Example:** `Device(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 1996 duration 7200`	Enters service-family configuration mode and sets the time period during which the authentication key in a key chain is received as valid.
Step 15	**send-lifetime** *start-time* [**local** {**duration** *seconds* | **end-time** | **infinite**}] **Example:** `Device(config-keychain-key)# send-lifetime 14:00:00 Jan 25 1996 duration 3600`	Configures a time period during which an authentication key on a key chain is valid to be sent.
Step 16	**exit** **Example:** `Device(config-keychain-key)# end`	Exits service-family interface configuration mode.

Configuring Logs for Neighbor Changes and Warnings

By default, the system logs neighbor adjacency changes to help you monitor the stability of the routing system and detect problems. If you disabled logging of such changes and want to re-enable logging, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **eigrp log-neighbor-changes**
6. **eigrp log-neighbor-warnings** *seconds*
7. **exit-service-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**eigrp log-neighbor-changes** **Example:** `Device(config-router-sf)# eigrp log-neighbor-changes`	Enables the logging of changes in EIGRP service-family neighbor adjacencies.

	Command or Action	Purpose
Step 6	**eigrp log-neighbor-warnings** *seconds* **Example:** `Router(config-router-sf)# eigrp log-neighbor-warnings 60`	Enables the logging of changes in service-family warning messages.
Step 7	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

Configuring the Percentage of Link Bandwidth Used for Cisco SAF

By default, packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth interface** configuration command. You may want to change the value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations). Use the following commands to configure the percentage of link bandwidth used for Cisco SAF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **bandwidth-percent** *maximum-bandwidth-percentage*
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system on the device.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet0/0`	Enables service-family interface configuration mode for the specified interface.
Step 6	**bandwidth-percent** *maximum-bandwidth-percentage* **Example:** `Device(config-router-sf-interface)# bandwidth-percent 75`	Configures the maximum percentage of bandwidth used by the link for Cisco SAF.
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Setting Metric Dampening Intervals for Cisco SAF Interfaces

Because metric components can be changed rapidly, the frequency of the changes can have an impact on the network. Frequent changes require that prefixes learned though the SAF interface be updated and sent to all adjacencies. This update can result in further updates and, in a worst-case scenario, cause network-wide churn. To prevent such effects, metrics can be dampened or thresholds set so that any change that does not exceed the dampening threshold is ignored.

Network changes that cause an immediate update include when a device selects a new nexthop or a down interface or device.

Thresholds can be configured based on a change or on a time interval. If the dampening method is:

- Change-based, changes in routes learned though a specific interface or in the metrics for a specific interface will not be advertised to adjacencies until the *computed* metric changes from the last advertised value are significant enough to cause an update to be sent.
- Interval-based, changes in routes learned though a specific interface or in the metrics for a specific interface will not be advertised to adjacencies until the *specified* interval is met or unless the change results in a new route path selection. When the timer expires, routes that have outstanding changes to report are sent. If a route changes and the final metric of the route matches the last updated metric, no updated routes are sent.

Refer to the following sections for information on configuring change-based and interval-based metric dampening parameters.

- Change-based Dampening Configuration, page 22
- Interval-based Dampening Configuration, page 23

Change-based Dampening Configuration

Use the following commands to set the maximum change-based dampening percentage for Cisco SAF interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **dampening-change** [*change-percentage*]
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet0/0`	Enables service-family interface configuration mode for the specified interface on the device.
Step 6	**dampening-change** [*change-percentage*] **Example:** `Device(config-router-sf-interface)# dampening-change 50`	Configures the percentage of change in a route learned through an EIGRP service-family interface that causes an update to be advertised to adjacent peers.
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Interval-based Dampening Configuration

Use the following commands to configure interval-based dampening for Cisco SAF interfaces. The value that you configure sets the interval when updates occur for topology changes that affect Cisco SAF interfaces and peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-name interface-number*
6. **dampening-interval** [*interval*]
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-name interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet0/0`	Enables service-family interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 6	**dampening-interval** [*interval*] **Example:** `Device(config-router-sf-interface)# dampening-interval 30`	Sets the EIGRP interval-based dampening interval (in seconds).
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Adjusting the Interval Between Hello Packets and the Hold Time

Routing devices periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. This information is used to discover neighbors and to learn when neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast multiaccess (NBMA) media on which the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower as specified in the **bandwidth interface** configuration command. The default hello interval remains at 5 seconds for high-speed NBMA networks. Note that for the purposes of Frame Relay and Switched Multimegabit Data Service (SMDS), networks may or may not be considered to be NBMA. These networks are considered NBMA if the interface has not been configured to use physical multicasting; otherwise they are not considered NBMA.

The hold time is advertised in hello packets and indicates to neighbors the length of time for which they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds. For slow-speed NBMA networks, the default hold time is 180 seconds. On congested and large networks, the default hold time might not be sufficient time for all devices to receive hello packets from their neighbors. In this case, you may want to increase the hold time. Do not adjust the hold time without advising your technical support personnel. To change the hold time on a specific interface for a particular routing process designated by an autonomous system number, use the **hold time** command.

You can adjust the interval between hello packets and the hold time. To change the interval between hello packets and the hold time, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **hello-interval** *seconds*
7. **hold-time** *seconds*
8. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet0/0`	Enables service-family interface configuration mode for the specified interface.
Step 6	**hello-interval** *seconds* **Example:** `Device(config-router-sf-interface)# hello-interval 50`	Configures the time period for an EIGRP service-family process.
Step 7	**hold-time** *seconds* **Example:** `Device(config-router-sf-interface)# hold-time 50`	Configures the time period for an EIGRP service-family routing process designated by an autonomous system number.

	Command or Action	Purpose
Step 8	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Disabling Split Horizon

When split horizon is enabled on an interface, it blocks route information (such as update and query packets) from being advertised by a device out of any interface from which that information originates. Controlling update and query packets in this manner reduces the possibility of routing loops.

This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is less than ideal. For these situations, including networks in which you have Cisco SAF configured, you may want to disable split horizon.

By default, split horizon is enabled on all interfaces. To disable split horizon, use the **no split-horizon** command in interface configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **no split-horizon**
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet0/0`	Enables service-family interface configuration mode for the specified interface.
Step 6	**no split-horizon** **Example:** `Device(config-router-sf-interface)# no split-horizon`	Disables split horizon.
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Setting Metric Maximum Hops

The maximum number of hops limits the number of hops a service can propagate to advertise its service. The default number of maximum hops is 100.

To limit the number of hops used to advertise a service, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **sf-interface** *interface-type interface-number*
6. **metric maximum-hops** *hop-count*
7. **exit-sf-interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enables a Cisco SAF service family for the specified autonomous system.
Step 5	**sf-interface** *interface-type interface-number* **Example:** `Device(config-router-sf)# sf-interface ethernet 0/0`	Enables service-family interface configuration mode for the specified interface on the device.

	Command or Action	Purpose
Step 6	**metric maximum-hops** *hop-count* **Example:** `Device(config-router-sf-interface)# metric maximum-hops 5`	Specifies a hop count for the routes that the IP routing software advertises as unreachable.
Step 7	**exit-sf-interface** **Example:** `Device(config-router-sf-interface)# exit-sf-interface`	Exits service-family interface configuration mode.

Configuring a Cisco SAF External Client

To configure a Cisco SAF external client, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology base**
6. **external-client** *client-label*
7. **exit-sf-topology**
8. **exit-service-family**
9. **exit**
10. **service-family external-client listen** {**ipv4** | **ipv6**} *tcp-port-number*
11. **external-client** *client-label* **basename**
12. **username** *user-name*
13. **password** *password-name*
14. **keepalive** *number*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enters service-family configuration mode and enables a Cisco SAF service family for the specified autonomous system.
Step 5	**topology base** **Example:** `Device(config-router-sf)# topology base`	Enables service-family interface topology configuration mode.
Step 6	**external-client** *client-label* **Example:** `Device(config-router-topology)# external-client example`	Configures a Cisco SAF external client with the specified client label.
Step 7	**exit-sf-topology** **Example:** `Device(config-router-sf-topology)# exit-sf-topology`	Exits service-family interface topology configuration mode.
Step 8	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 10	**service-family external-client listen** {**ipv4** | **ipv6**} *tcp-port-number* **Example:** `Device(config)# service-family external-client listen ipv4 5050`	Configures a Cisco SAF external client TCP port to use to communicate with a Cisco SAF forwarder. The valid port range is 1024 to 65536.
Step 11	**external-client** *client-label* **basename** **Example:** `Device(config-external-client)# external-client example basename`	Configures a Cisco SAF external client with the specified client label and a basename. Specifying the **basename** keyword allows SAF external clients to use a naming convention based on the client label. The naming convention takes the form of *client-label* @[1-1024] where you can specify a maximum of 1024 SAF external clients. For example, if the **external-client** command specifies a client label of *example* , then the basename for a SAF external client would be *example@1*. Another SAF external client would be *example@2* , and so on up to a maximum of 1024 basenames (*@1024*).
Step 12	**username** *user-name* **Example:** `Device(config-external-client)# username example`	Enables external-client label configuration mode and configures a Cisco SAF external client with the specified username.
Step 13	**password** *password-name* **Example:** `Device(config-external-client-mode)# password examplepass`	Configures a password for a Cisco SAF external client. The minimum password length is 11 characters.
Step 14	**keepalive** *number* **Example:** `Device(config-external-client-mode)# keepalive 360000`	(Optional) Specifies the keepalive timer for the Cisco SAF external client. The keepalive value is in milliseconds. The default is 9600 ms.

	Command or Action	Purpose
Step 15	**end** **Example:** `Device(config-external-client-mode)# end`	Exits external-client label configuration mode.

Displaying Cisco SAF Statistics

To display Cisco SAF statistics, use the following commands in privileged EXEC mode.

SUMMARY STEPS

1. **show service-routing xmcp clients** [*ip-address* | *handle*] [**detail**]
2. **show service-routing xmcp server**
3. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **accounting**
4. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **clients** [**detail**]
5. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **events** [*starting-event-number ending-event-number*]
6. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **interfaces** [*interface-type interface-number*] [**detail**]
7. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **subscriptions**
8. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **timers**
9. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **summary**
10. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **zero successors**
11. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology**
12. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology active**
13. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology all-links**
14. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology base** *service-instance-number* **clients** [**detail**]
15. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology** [**detail-links**]
16. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology events** [*starting-event-number ending-event-number*]
17. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology pending**
18. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology** [**service-type connected** | **external** | **internal** | **local** | **redistributed** | **summary**]
19. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology sia-events**[*starting-event-number ending-event-number*]
20. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology sia-statistics** [*ip-address*]
21. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology summary**
22. **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology zero-successors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show service-routing xmcp clients** [*ip-address* | *handle*] [**detail**] **Example:** `Device# show service-routing xmcp clients detail`	Displays information about connected XMCP clients.

	Command or Action	Purpose
Step 2	**show service-routing xmcp server** **Example:** `Device# show service-routing xmcp server`	Displays information about clients, external clients, or subscriptions configured for Cisco SAF.
Step 3	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **accounting** **Example:** `Device# show eigrp service-family ipv4 4453 accounting`	Displays accounting information about Cisco SAF.
Step 4	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **clients** [**detail**] **Example:** `Device# show eigrp service-family ipv4 4453 clients`	Displays information about Cisco SAF Clients.
Step 5	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **events** [*starting-event-number ending-event-number*] **Example:** `Device# show eigrp service-family ipv4 4453 events`	Displays information about Cisco SAF events.
Step 6	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **interfaces** [*interface-type interface-number*] [**detail**] **Example:** `Device# show eigrp service-family ipv4 4453 interfaces`	Displays information about Cisco SAF interfaces.
Step 7	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **subscriptions** **Example:** `Device# show eigrp service-family ipv4 4453 subscriptions`	Displays information about Cisco SAF subscriptions.

	Command or Action	Purpose
Step 8	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **timers** **Example:** `Device# show eigrp service-family ipv4 4453 timers`	Displays information about Cisco SAF timers.
Step 9	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **summary** **Example:** `Device# show eigrp service-family ipv4 4453 summary`	Displays summary information about Cisco SAF.
Step 10	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **zero successors** **Example:** `Device# show eigrp service-family ipv4 4453 zero successors`	Displays information about Cisco SAF zero successors.
Step 11	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology** **Example:** `Device# show eigrp service-family ipv4 4453 topology`	Displays information about the Cisco SAF topology table.
Step 12	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology active** **Example:** `Device# show eigrp service-family ipv4 4453 topology active`	Displays only active entries for a Cisco SAF topology table.
Step 13	**show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology all-links** **Example:** `Device# show eigrp service-family ipv4 4453 topology all-links`	Displays all active link entries for a Cisco SAF topology table.

Command or Action	Purpose
Step 14 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology base** *service-instance-number* **clients** [**detail**] **Example:** `Device# show eigrp service-family ipv4 4453 topology base clients detail`	Displays all active link entries for a Cisco SAF topology base.
Step 15 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology** [**detail-links**] **Example:** `Device# show eigrp service-family ipv4 4453 topology detail-links`	Specifies all links in the topology table.
Step 16 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology events** [*starting-event-number ending-event-number*] **Example:** `Device# show eigrp service-family ipv4 4453 topology events`	Specifies all events in the topology table.
Step 17 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology pending** **Example:** `Device# show eigrp service-family ipv4 4453 topology pending`	Displays all active entries in the topology table that are waiting either for an update or reply from a neighbor.
Step 18 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology** [**service-type connected** | **external** | **internal** | **local** | **redistributed** | **summary**] **Example:** `Device# show eigrp service-family ipv4 4453 topology service-type connected`	Displays information about the specified service type for a Cisco SAF topology table.
Step 19 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology sia-events**[*starting-event-number ending-event-number*] **Example:** `Device# show eigrp service-family ipv4 4453 topology sia-events`	Displays logged Stuck in Active (SIA) events in the Cisco SAF topology table.

Command or Action	Purpose
Step 20 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology sia-statistics** [*ip-address*] **Example:** `Device# show eigrp service-family ipv4 4453 topology sia-statistics 10.10.10.1`	Displays Stuck in Active (SIA) events for a Cisco SAF topology table.
Step 21 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology summary** **Example:** `Device# show eigrp service-family ipv4 4453 topology summary`	Displays a summary of a Cisco SAF topology table.
Step 22 **show eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **topology zero-successors** **Example:** `Device# show eigrp service-family ipv4 4453 topology zero-successors`	Displays information about available services that have zero successors in a Cisco SAF topology table.

Deleting Information from a Cisco SAF Configuration

To delete service-family information from a Cisco SAF configuration, use the following commands in EXEC mode.

SUMMARY STEPS

1. **clear service-family xmcp client** {*ip-address* | *handle*}
2. **clear eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number*
3. **clear eigrp service-family neighbors** *neighbor-address* | *interface-type interface-number*

DETAILED STEPS

Command or Action	Purpose
Step 1 **clear service-family xmcp client** {*ip-address* | *handle*} **Example:** `Device# clear service-family xmcp client 1.1.1.1`	Disconnects a connected XMCP client.

	Command or Action	Purpose
Step 2	**clear eigrp service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] *autonomous-system-number* **Example:** `Device# clear eigrp service-family ipv4 4453`	Deletes neighbors formed using the IPv4 or IPv6 protocol family for the specified autonomous system. Optionally, you can delete all virtual routing forwarding (VRF) instance tables or a specific VRF table for an IP address. **Note** Using the **clear eigrp service-family ipv6** command requires an IPv6-enabled SAF client, which currently does not exist.
Step 3	**clear eigrp service-family neighbors** *neighbor-address* | *interface-type interface-number* **Example:** `Device# clear eigrp service-family neighbors Ethernet 0/0`	Deletes neighbors formed using the IPv4 protocol family from the neighbor table. Optionally, you can resynchronize with a peer without an adjacency reset (soft). Optionally, you can delete the interface type and number from the neighbor table that contains all entries learned through this interface.

Configuration Examples for Cisco SAF

Example: Enabling Cisco SAF

The following example enters router configuration mode, configures a Cisco SAF forwarder, enables the service-family forwarder process, and configures an autonomous system named 4533.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
```

Examples: Configuring Cisco SAF Interfaces

The following example places the device in service-family configuration mode and enables all interfaces.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# sf-interface default
Device(config-router-sf-interface)# no shutdown
```

The following example places the device in service-family configuration mode and enables Ethernet interface 0/0.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# sf-interface ethernet0/0
```

The following example places the device in service-family configuration mode and enables SAF on all interfaces, except Ethernet0/0 interface.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 3
Device(config-router-sf)# interface default
Device(config-router-sf)# sf-interface ethernet0/0
Device(config-router-sf-interface)# shutdown
Device(config-router-sf-interface)# end
```

The following example places the device in service-family configuration mode, enables SAF on Ethernet interfaces 2/0 and 2/1, and disables SAF on all other interfaces.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 2
Device(config-router-sf)# sf-interface default
Device(config-router-sf-interface)# shutdown
Device(config-router-sf-interface)# sf-interface ethernet2/0
Device(config-router-sf-interface)# no shutdown
Device(config-router-sf-interface)# sf-interface ethernet2/1
Device(config-router-sf-interface)# no shutdown
Device(config-router-sf-interface)# end
```

Example: Configuring Cisco SAF Topology

The following examples configures a Cisco SAF topology base.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# sf-interface default
Device(config-router-sf-interface)# no shutdown
Device(config-router-sf-interface)# topology base
```

Example: Configuring Cisco SAF Stub Routing

The following examples configures a Cisco SAF forwarder as a stub device.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# eigrp stub connected
```

Example: Configuring Cisco SAF with IP-RIP

The following configuration example enables Cisco SAF with IP RIP routing on network 10.0.0.0.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# exit-service-family
Device(config-router)# router rip
Device(config-router)# network 10.0.0.0
```

Example: Configuring Cisco SAF with OSPF

The following configuration example enables Cisco SAF with OSPF routing on network 10.0.0.0, area 0.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# exit-service-family
Device(config-router)# router ospf 787
Device(config-router)# network 10.0.0.0  0.0.0.255 area 0
```

Example: Configuring Cisco SAF with EIGRP

The following configuration example enables Cisco SAF with EIGRP routing on network 10.0.0.0.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 6476
Device(config-router-sf)# network 10.0.0.0  0.0.0.255
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# exit-service-family
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# topology base
```

Note There is no requirement to run routing over the same interfaces or networks in which services are distributed; however this could lead to services being distributed to areas where reachability is not guaranteed.

Example: Configuring Cisco SAF Forwarders Located on Separate LANs

The following example configures two Cisco SAF forwarders located on separate LANs.

Note Use loopback mode to configure remote neighbors.

Cisco SAF Forwarder 1

```
Device(config)# interface loopback1
Device(config-if)# ip address 10.1.1.1 255.255.255.255
Device(config-if)# exit
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 1
Device(config-router-sf)# neighbor 10.2.2.2 loopback1 remote 10
```

Cisco SAF Forwarder 2

```
Device(config)# interface loopback1
Device(config-if)# ip address 10.2.2.2 255.255.255.255
Device(config-if)# exit
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 1
Device(config-router-sf)# neighbor 10.1.1.1 loopback1 remote 10
```

Note This example assumes that IP routing is configured between the two devices and that the devices can ping both loopbacks.

Example: Configuring a Centralized Cisco SAF Forwarder

The following example configures a centralized Cisco SAF forwarder that sends all service advertisements to neighbors on IP addresses 10.4.15.5 and 10.4.15.1.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# sf-interface loopback0
Device(config-router-sf-interface)# no split-horizon
Device(config-router-sf-interface)# exit-sf-interface
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# neighbor 10.4.15.5 Loopback0 remote 20
Device(config-router-sf)# neighbor 10.4.15.1 Loopback0 remote 20
Device(config-router-sf)# exit-service-family
```

Examples: Configuring a Cisco SAF Client

The following example configures a Cisco SAF external client named *example*, with a username of *username_example*, a password of *password_example*, and a keepalive setting of 360,000 seconds.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# external-client example
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# exit-service-family
Device(config-router)# exit
Device(config)# service-family external-client listen ipv4 3444
Device(config-external-client)# external-client example
Device(config-external-client-mode)# username username_example
Device(config-external-client-mode)# password password_example
Device(config-external-client-mode)# keepalive 360000
```

The following example configures five Cisco SAF external clients named *example1* through *example5*, with usernames of *username_example1* through *username_example5*, passwords of *password_example1* through *password_example5*, and keepalive settings of 360,000 seconds.

```
Device(config)# router eigrp saf
Device(config-router)# service-family ipv4 autonomous-system 4533
Device(config-router-sf)# topology base
Device(config-router-sf-topology)# external-client example1
Device(config-router-sf-topology)# external-client example2
Device(config-router-sf-topology)# external-client example3
Device(config-router-sf-topology)# external-client example4
Device(config-router-sf-topology)# external-client example5
Device(config-router-sf-topology)# exit-sf-topology
Device(config-router-sf)# exit-service-family
Device(config-router)# exit
Device(config)# service-family external-client listen ipv4 3444
Device(config-external-client)# external-client example1
Device(config-external-client-mode)# username username_example1
Device(config-external-client-mode)# password password_example1
Device(config-external-client-mode)# keepalive 360000
Device(config-external-client-mode)# external-client example2
Device(config-external-client-mode)# username username_example2
Device(config-external-client-mode)# password password_example2
Device(config-external-client-mode)# keepalive 360000
```

```
Device(config-external-client-mode)# external-client example3
Device(config-external-client-mode)# username username_example3
Device(config-external-client-mode)# password password_example3
Device(config-external-client-mode)# keepalive 360000
Device(config-external-client-mode)# external-client example4
Device(config-external-client-mode)# username username_example4
Device(config-external-client-mode)# password password_example4
Device(config-external-client-mode)# keepalive 360000
Device(config-external-client-mode)# external-client example5
Device(config-external-client-mode)# username username_example5
Device(config-external-client-mode)# password password_example5
Device(config-external-client-mode)# keepalive 360000
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Service Advertisement Framework commands	*Cisco IOS Service Advertisement Framework Technology Command Reference*
Cisco EIGRP stub routing	*Configuring EIGRP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco SAF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Cisco Service Advertisement Framework

Feature Name	Software Releases	Feature Configuration Information
Cisco Service Advertisement Framework	12.2(33)SRE 12.2(33)SXI4 12.2(33)XNE 12.2(50)SY 15.0M 15.0(1)S 15.1(1)SG 15.1(2)S 15.1T 15.2(1)T 15.2(2)S 15.2(3)T Cisco IOS XE Release 2.5 Cisco IOS XE Release 3S, Cisco IOS XE Release 3.3SG Cisco IOS XE Release 3.4S Cisco IOS XE Release 3.6S	This feature allows applications to discover the existence, location, and configuration of networked resources within networks, and it provides a timely and reliable awareness of the services within networks, as applications advertise and discover services on networks. This feature was introduced in Cisco IOS Release 15.0M. In Cisco IOS XE Release 2.5, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced in this feature: • **accept-lifetime** • **authentication (service-family)** • **bandwidth-percent** • **clear eigrp service-family** • **dampening-change** • **dampening-interval** • **debug eigrp service-family** • **default external-client** • **default (SAF)** • **default-metric (EIGRP)** • **eigrp log-neighbor-changes** • **eigrp-log-neighbor-warnings** • **eigrp router-id** • **eigrp stub (service-family)** • **exit-service-family** • **exit-sf-interface** • **exit-sf-topology** • **external-client** • **hello-interval** • **hold-time** • **keepalive (SAF)** • **key** • **key chain**

Feature Name	Software Releases	Feature Configuration Information
		• **key-string (authentication)** • **maximum-service (EIGRP)** • **metric weights (EIGRP)** • **neighbors (service-family)** • **next-hop-self** • **password (SAF)** • **send-lifetime** • **service-family** • **service-family external-client listen** • **sf-interface** • **show eigrp service-family** • **show eigrp service-family ipv4 topology** • **show eigrp service-family ipv6 topology** • **show eigrp tech-support** • **shutdown** • **split-horizon** • **timers** • **topology** • **username (SAF)**

Configuring Dynamic Neighbors

When neighbors are not adjacent, normal Cisco Service Advertisement Framework (SAF) peering mechanisms cannot be used to exchange SAF information over the networking cloud. The neighbors are often multiple hops away, and separated by dark nets (routers not running SAF).

To support this type of network, SAF provides the **neighbor** command, which allows remote neighbors to be configured and sessions established though unicast packet transmission. However, as the number of forwarders needing to exchange SAF information over the networking cloud increases, unicast SAF neighbor definitions may become cumbersome to manage. Each neighbor must be manually configured, resulting in increased operational costs.

To better accommodate deployment of these topologies, ease configuration management, and reduce operational costs, the Dynamic Neighbors feature provides support for the dynamic discovery of remote unicast and multicast neighbors (referred to as “remote neighbors”). Remote neighbor support allows Cisco SAF peering to one or more remote neighbors, which may not be known at the time the device is configured, thus reducing configuration management.

This section contains the following major topics:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Dynamic Neighbors

Before configuring SAF dynamic neighbors, ensure that when using:

- Unicast-listen mode—IP connectivity (reachability) exists between devices that need to do dynamic peering.
- Multicast-group mode—Multicast is running on the network.

Restrictions for Dynamic Neighbors

- The **remote-neighbors** command requires a loopback as a source interface.
- Only named access control lists (ACLs) are permitted with the **allow-list** keyword. Numbered ACLs are not permitted.

Within a service family, the following restrictions apply:

- Only one **remote-neighbors unicast-listen** command and one **remote-neighbors multicast-group** command may be configured per interface. For example, you cannot configure **remote-neighbors source Loopback1 multicast-group 224.1.1.1** and **remote-neighbors source Loopback1 multicast-group 224.1.1.2**. If you want to configure multiple different multicast-group addresses in the same service-family, you must use multiple source interfaces.
- A multicast-group address may be associated to a single source interface. For example, you cannot configure **remote-neighbors source Loopback1 multicast-group 224.1.1.1** and **remote-neighbors source Loopback2 multicast-group 224.1.1.1**.

Information About Dynamic Neighbors

Dynamic Neighbors Overview

When neighbors are not adjacent, normal Cisco SAF peering mechanisms cannot be used to exchange SAF information over the networking cloud. The neighbors are often multiple hops away, and separated by dark nets (devices not running SAF).

To support this type of network, SAF provides the **neighbor** command, which allows remote neighbors to be configured and sessions established through unicast packet transmission. However, as the number of Forwarders needing to exchange SAF information over the networking cloud increases, unicast SAF neighbor definitions may become cumbersome to manage. Each neighbor must be manually configured, resulting in increased operational costs.

To better accommodate deployment of these topologies, ease configuration management, and reduce operational costs, the Dynamic Neighbors feature provides support for the dynamic discovery of remote unicast and multicast neighbors (referred to as “remote neighbors”). Remote neighbor support allows Cisco SAF peering to one or more remote neighbors, which may not be known at the time the device is configured, thus reducing configuration management.

Remote Neighbor Session Policy

When using remote unicast-listen or remote multicast-group neighbor configurations, SAF neighbor IP addresses are not predefined, and neighbors may be many hops away. A device with this configuration could peer with any device that sends a valid HELLO packet. Because of security considerations, this open aspect requires policy capabilities to limit peering to valid devices and to restrict the number of neighbors in order to limit resource consumption. This capability is accomplished using the following manually configured parameters, and takes effect immediately.

Neighbor Filter List

The optional **allow-list** keyword, available in the **remote-neighbors** command, enables you to use an access list (access control list) to specify the remote IP addresses from which Cisco SAF neighbor connections may be accepted. If you do not use the **allow-list** keyword, then all IP addresses (permit any) will be accepted.

The access control list (ACL) defines a range of IPv4 or IPv6 IP addresses with the following conditions:

- Any neighbor that has a source IP address that matches an IP address in the access list will be allowed (or denied) based on the user configuration.
- If the **allow-list** keyword is not specified, any IP address will be permitted (permit any).
- The **allow-list** keyword is supported only for remote multicast-group and unicast-listen neighbors. It is not available for static, remote static, or local neighbors.
- Incoming Cisco SAF packets that do not match the specified access list will be rejected.

Maximum Remote Neighbors

The optional **max-neighbors** keyword, available in the **remote-neighbors** command, enables you to specify a maximum number of remote neighbors that Cisco SAF can create using the remote neighbor configurations. When the maximum number of remote neighbors has been created for a configuration, Cisco SAF rejects all subsequent connection attempts for that configuration. This option helps to protect against denial-of-service attacks that attempt to create many remote neighbors in an attempt to overwhelm device resources.

The **max-neighbors** configuration option has the following conditions:

- This option is supported only for remote multicast-group or unicast-listen neighbors. It is not available for local, static, or remote static neighbors.
- There is no default maximum. If you do not specify a maximum number of remote neighbors, the number of remote neighbors is limited only by available memory and bandwidth.
- Reducing the maximum number of remote neighbors to less than the current number of sessions will result in the neighbors (in no specific order) being dropped until the count reaches the new limit.

Configuration Changes for the Neighbor Filter List and Maximum Number of Remote Neighbors

When the **allow-list** or **max-neighbors** configurations are changed, any existing remote Cisco SAF sessions that are no longer allowed by the new configuration will be removed automatically and immediately. Pre-existing neighbors that are still allowed by the new configuration will not be affected.

Neighbor Types

The following terms are used when describing neighbor types:

- local neighbor—A neighbor that is adjacent on a shared subnet (or common subnet) and uses a link-local multicast address for packet exchange. This is the default type of neighbor in Cisco SAF.
- static neighbor—Any neighbor that uses unicast to communicate, is one hop away, is on a common subnet, and whose IP address has been specified using the **neighbor** *ip-address* command.
- remote neighbor—Any neighbor that is multiple hops away, including Remote Static Neighbors.
- remote static neighbor--Any neighbor that uses unicast to communicate, is multiple hops away, and whose IP address has been specified using the **neighbor** *ip-address* command.
- remote multicast-group—Any neighbor that is multiple hops away, but does not have its IP address manually configured using the **neighbor** *ip-address* command, and uses a configured multicast group address for packet exchange.
- remote unicast-listen (or simply unicast-listen)—Any neighbor that uses unicast to communicate, is multiple hops away, and whose IP address has not been configured using the **neighbor** *ip-address* command.

Remote Unicast-Listen (Point-to-Point) Neighbors

For configurations in which multiple remote neighbors peer with a single hub (point-to-point), the hub can be configured for remote unicast-listen peering using the **remote-neighbors** command to allow the remote neighbors to peer with the hub without having to manually configure the remote neighbor IP addresses on the hub.

When configured with this command, the hub device:

- Uses its interface IP address as the source IP address for any unicast transmissions. This IP address must be routable.
- Requires neighbors that peer with the hub to be configured using the **neighbor** *ip-address* **loopback** *loopback-interface-number* **remote** *maximum-hops* command where *ip-address* is the unicast address of the local device interface.
- Listens for unicast HELLO packets on the interface specified in the **remote-neighbor** command.
- Accepts a unicast HELLO packet if it is in the IP address range configured using the **allow-list** keyword, or any unicast HELLO packet if an allow list is not defined.
- Rejects multicast HELLO packets from any neighbor that is also sending unicast HELLO packets and is permitted by the unicast allow list (or all neighbors if an allow list is not defined).
- Begins normal neighbor establishment using the IP addresses of the remote neighbors for packet transmission once the neighbor relationship is established.

Remote Multicast-Group (Multipoint-to-Multipoint) Neighbors

Multicast can be used to provide an efficient transport between multiple Cisco SAF neighbors. A single multicast-group address can be used for multiple Cisco SAF neighbors to exchange information within the same multicast group. To configure multipoint-to-multipoint configurations, use the **multicast-group** keyword available in the **remote neighbors** command.

When configured with this command, the device:

- Uses the interface IP address as the source IP address for any unicast transmissions. This IP address must be routable.
- Uses the configured multicast-group address for all multicast packets sent and received.
- Requires all forwarders and routers, which form the multipoint-to-multipoint neighbor relationships, to be configured using the same multicast-group IP address.
- Requires multicast forwarding for the defined multicast-group address to be configured and functional for packet delivery.

Inheritance and Precedence of the Remote Neighbor Configurations

Static neighbors configured with the **neighbor** *ip-address* or the **neighbor** *ip-address* **remote** commands take precedence over the remote neighbors that are created as a result of the **remote-neighbors** command. If the remote IP address of an incoming unicast Cisco SAF connection matches both a static neighbor and the remote unicast-listen neighbor access list, the static neighbor is used and no remote unicast-listen neighbor is created. If you configure a new static neighbor while a remote neighbor for the same remote IP address already exists, Cisco SAF automatically removes the remote unicast-listen neighbor.

Remote unicast-listen neighbors take precedence over remote multicast-group neighbors. If Cisco SAF is receiving both unicast and multicast HELLOs from the same remote IP address targeted at the same local interface, the neighbor will be treated as unicast (unicast-listen) rather than multicast (multicast-group) for packet exchange.

How to Configure Dynamic Neighbors

- Configuring Dynamic Neighbors, page 51

Configuring Dynamic Neighbors

To configure Cisco SAF dynamic neighbors, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **remote-neighbors source** *interface* {**unicast-listen** | **multicast-group** *group-address* } [**allow-list** *access-list-name*] [**max-neighbors** *max-remote-peers*]
6. **exit-service-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp saf`	Enables an EIGRP virtual instance.
Step 4	**service-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# service-family ipv4 autonomous-system 4453`	Enables a Cisco SAF service family for the specified autonomous system.
Step 5	**remote-neighbors source** *interface* {**unicast-listen** | **multicast-group** *group-address* } [**allow-list** *access-list-name*] [**max-neighbors** *max-remote-peers*] **Example:** `Device(config-router-sf)# remote-neighbors source Loopback1 unicast-listen allow-list myNeighborList`	Configures a SAF process that enables remote neighbors to accept inbound connections from any remote IP address. • **allow-list** keyword to use an access list (access control list) to specify the remote IP addresses from which Cisco SAF neighbor connections may be accepted. If you do not use the allow list keyword, all IP addresses will be accepted. • **max-neighbors** keyword to specify the maximum number of remote neighbors. If you do not specify a number, the maximum number of remote neighbors is limited only by available memory and bandwidth.
Step 6	**exit-service-family** **Example:** `Device(config-router-sf)# exit-service-family`	Exits service-family configuration mode.

Configuration Examples for Dynamic Neighbors

Examples: Configuring Cisco SAF Dynamic Neighbors

The following examples show how to configure both devices involved in the neighbor relationship.

This example uses the **unicast-listen** keyword to configure remote neighbors to accept inbound connections from IP addresses that match the access list myNeighborList.

```
Device1(config)# interface Loopback1
Device1(config-if)# ip address 10.1.1.1 255.255.255.255
Device1(config-if)# exit
Device1(config)# ip access-list standard myNeighborList
Device1(config-std-nacl)# permit 10.0.0.0 0.255.255.255
Device1(config-std-nacl)# exit
Device1(config)# router eigrp virtual-name
Device1(config-router)# service-family ipv4 autonomous-system 4453
Device1(config-router-sf)# remote-neighbors source Loopback1 unicast-listen allow-list
myNeighborList

Device2(config)# interface Loopback2
Device2(config-if)# ip address 10.2.2.2 255.255.255.255
Device2(config-if)# exit
Device2(config)# router eigrp virtual-name
Device2(config-router)# service-family ipv4 autonomous-system 4453
Device2(config-router-sf)# neighbor 10.1.1.1 Loopback2 remote 20
```

This example uses the **multicast-group** keyword to use IP multicast to discover remote neighbors and form remote neighbor relationships. It also specifies 30 as the maximum number of inbound connections from remote neighbors that a member of the multicast group may accept.

```
Device1(config)# interface Loopback1
Device1(config-if)# ip address 10.1.1.1 255.255.255.255
Device1(config-if)# ip pim sparse-mode
Device1(config-if)# exit
Device1(config)# router eigrp virtual-name
Device1(config-router)# service-family ipv4 autonomous-system 4453

Device1(config-router-sf)# remote-neighbors source Loopback1 multicast-group 224.44.56.1
max-neighbors 30
Device2(config)# interface Loopback2
Device2(config-if)# ip address 10.2.2.2 255.255.255.255
Device2(config-if)# ip pim sparse-mode
Device2(config-if)# exit
Device2(config)# router eigrp virtual-name
Device2(config-router)# service-family ipv4 autonomous-system 4453
Device2(config-router-sf)# remote-neighbors source Loopback2 multicast-group 224.44.56.1
max-neighbors 30
```

Example: Enhancements to the show neighbor detail Command

The existing detail option of the show neighbor command will be extended to show the information as to how the neighbor was configured. If the neighbor is:

- Configured as static single hop. It is listed as a static neighbor.
- Configured as static multi-hop. It is listed as a remote static neighbor (static multi-hop).
- Configured as remote unicast-listen. It is listed as a remote neighbor (unicast-listen).

- Configured as remote multicast-group. It is listed as a group neighbor (multicast-group <ip addr>.)
- Not manually configured. It has no special listing here.

For example:

```
Device# show eigrp ipv4 neighbor detail

EIGRP-SFv4 VR(test) Service-Family Neighbors for AS(100)
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num

1 10.1.2.1 Lo0 12 00:02:20 3 100 0 3
Static neighbor
Version 8.0/4.0, Retrans: 0, Retries: 0
Topology-ids from peer - 0

1 1.1.2.1 Lo0 12 00:02:20 3 100 0 3
Remote Static neighbor (static multihop)
Version 8.0/4.0, Retrans: 0, Retries: 0
Topology-ids from peer - 0

1 10.1.1.4 Lo0 12 00:02:01 3 100 0 1
Remote neighbor (unicast-listen)
Version 8.0/4.0, Retrans: 0, Retries: 0
Topology-ids from peer - 0

1 10.1.1.5 Lo0 14 00:04:07 1552 5000 0 3
Group neighbor (multicast-group 224.1.1.1)
Version 8.0/4.0, Retrans: 0, Retries: 0
Topology-ids from peer - 0

0 3.1.2.3 Et1/0 12 00:02:20 1999 5000 0 1
Version 8.0/4.0, Retrans: 0, Retries: 0
Topology-ids from peer - 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Service Advertisement Framework commands	*Cisco IOS Service Advertisement Framework Technology Command Reference*
Cisco EIGRP stub routing	*Configuring EIGRP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Dynamic Neighbors

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Dynamic Neighbors

Feature Name	Releases	Feature Information
Dynamic Neighbors	15.1(2)S 15.2(2)S 15.1(1)SG 15.1(1)SY 15.2(3)T Cisco IOS XE Release 3.6S Cisco IOS XE Release 3.3SG	The Dynamic Neighbors feature provides support for the dynamic discovery of remote unicast and multicast neighbors (referred to as “remote neighbors”). Remote neighbor support allows Cisco SAF peering to one or more remote neighbors. The following commands were introduced or modified: • **authentication mode** • **remote-neighbors source** • **show eigrp service-family external-client**

Configuring Capabilities Manager

Capabilities Manager is enabled by default at system startup. At startup, it registers as a Service Routing client and proceeds to discover various capabilities of the hardware and software platform.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Capabilities Manager

- To ensure that a device has Capabilities Manager available, enter the **show service-routing plugin capman** command.
  To enable a device to distribute its capabilities information, configure a SAF forwarder on the device.
- To view capabilities information present on other devices in the network, configure a SAF forwarder.

Information About Capabilities Manager

Capabilities Discovery

Capabilities Manager discovers only whether a capability is supported on the local system. It does not discover whether the capability is configured or enabled, nor does it discover any other information about the capability for other devices in the network.

Capabilities information will be installed into the local Network Information Base (NIB) as Service Routing data and will be made available for advertisement by any SAF forwarder to the Service Routing network. Capabilities information is passed to the Service Routing infrastructure in XML format and is stored in the local NIB.

Interoperability with SAF Forwarder

Capabilities Manager does not advertise capabilities information to the Service Routing network. A SAF forwarder distributes capabilities information. However, a SAF forwarder is not required in order for Capabilities Manager to function. If a SAF forwarder is not configured, the capabilities information is bound to the local device and is not distributed to other devices in the network. When a SAF forwarder is configured, it will distribute all capabilities information by default.

Capabilities Information

Capabilities information is installed in the Network Information Base (NIB) as Service Routing data. It is identified by a SAF address in the form of:

- Service ID—Capabilities Manager uses service ID 100.
- Subservice ID—Capability group ID. The subservice ID indicates the group ID of the capabilities data type.
- Instance number—Unique identifier for the local device. It is assigned in order of the hardware serial number, default MAC address, IPv4 device ID, or IPv6 device ID.

Capabilities Groups

Capabilities Manager classifies capabilities by group to facilitate query and retrieval, and assigns each group a unique ID. Capabilities Manager provides the following capability groups:

- 1 (HARDWARE)
- 2 (SOFTWARE)

- Hardware Group Information, page 58
- Software Group Information, page 59

Hardware Group Information

Hardware information is designated as group ID 1. Group 1 provides the following capabilities information, when available. All hardware information may not be available on each platform that supports Capabilities Manager.

- Hostname
- Platform
- Main memory size
- IO memory size

Software Group Information

Software information is designated as group ID 2. Group 2 provides the following capabilities information, when available. All software information may not be available on each platform that supports Capabilities Manager.

- Hostname
- Software
- Image
- Version
- Software subsystems:
  - IP Multicast
  - insp_appfw
  - ip_sla_responder
  - eigrp_ipv4
  - eigrp_ipv6
  - ospf
  - ospfv3
  - isis
  - isis_ipv6
  - bgp_ipv4
  - bgp_ipv6
  - service_routing

XML Schema for Capabilities Data

If you have an Extensible Messaging Client Protocol (XMCP) client (external client) connected to a SAF forwarder, you can subscribe to the Capabilities Manager, which is service ID 100. The data can be interpreted using the following XML schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:element name="Capabilities" type="CapabilitiesType" />
<xs:complexType name="CapabilitiesType">
<xs:sequence>
<xs:element ref="Group" minOccurs="1" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
<xs:element name="Group" type="GroupType" />
<xs:complexType name="GroupType">
<xs:sequence>
<xs:element ref="Capability" minOccurs="1" maxOccurs="unbounded" />
</xs:sequence>
<xs:attribute name="Name" type="xs:normalizedString" use="required" />
</xs:complexType>
<xs:element name="Capability" type="CapabilityType" />
<xs:complexType name="CapabilityType">
<xs:sequence>
<xs:element name="Value" type="xs:normalizedString" />
</xs:sequence>
<xs:attribute name="Name" type="xs:normalizedString" use="required" />
</xs:complexType>
</xs:schema>
Example:
<Capabilities>
<Group Name="HARDWARE">
<Capability Name="Hostname">
<Value>R100</Value>
```

```
</Capability>
<Capability Name="Platform">
<Value>Solaris Unix (Sparc) processor</Value>
</Capability>
<Capability Name="MainMemorySize">
<Value>63683Kbytes</Value>
</Capability>
</Group>
<Group Name="SOFTWARE">
<Capability Name="HostName">
<Value>R100</Value>
</Capability>
<Capability Name="Software">
<Value>Cisco IOS Software</Value>
</Capability>
<Capability Name="Image">
<Value>Solaris Software (UNIX-ADVENTERPRISE-M)</Value>
</Capability>
<Capability Name="Version">
<Value>Experimental Version 15.1(20110323:093227)</Value>
</Capability>
<Capability Name="ipmulticast">
<Value>Subsystem loaded</Value>
</Capability>
<Capability Name="eigrp_ipv4">
<Value>Subsystem loaded</Value>
</Capability>
</Group>
</Capabilities>
```

How to Configure Capabilities Manager

Disabling and Enabling Capabilities Manager

Capabilities Manager is enabled by default. You can disable and reenable Capabilities Manager at any time.

- Disabling Capabilities Manager will remove all the capabilities information that is installed in the local Network Information Base (NIB) and unregister the information from Service Routing.
- Reenabling Capabilities Manager will rediscover capabilities and provide information to the local NIB and to the Service Routing network.

Perform this task to disable and reenable Capabilities Manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no service-routing capabilities-manager**
4. **service-routing capabilities-manager**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no service-routing capabilities-manager** **Example:** `Device(config)# no service-routing capabilities-manager`	Disables Capabilities Manager.
Step 4	**service-routing capabilities-manager** **Example:** `Device(config)# service-routing capabilities-manager`	Enables Capabilities Manager.

Displaying Capabilities Manager Information

SUMMARY STEPS

1. **enable**
2. **show service-routing plugins capman**
3. **show service-routing capabilities-manager internal**
4. **show service-routing capabilities-manager** [**group** *value*] [**local**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show service-routing plugins capman** **Example:** `Device# show service-routing plugins capman`	Displays information about Capabilities Manager plugins.
Step 3	**show service-routing capabilities-manager internal** **Example:** `Device# show service-routing capabilities-manager internal`	Displays information about Capabilities Manager.
Step 4	**show service-routing capabilities-manager** [**group** *value*] [**local**] **Example:** `Device# show service-routing capabilities-manager group 1 local`	Displays information about registered capabilities.

Clearing Registered Capabilities Information

Perform this task to clear current capabilities information from the NIB. Once the NIB is cleared, Capabilities Manager will automatically rediscover new capabilities.

SUMMARY STEPS

1. **enable**
2. **clear service-routing capabilities-manager**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear service-routing capabilities-manager** **Example:** `Device# clear service-routing capabilities-manager`	Clears the current capabilities information from the NIB database. Capabilities Manager will automatically rediscover new capabilities.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Service Advertisement Framework commands	*Cisco IOS Service Advertisement Framework Technology Command Reference*
Cisco EIGRP stub routing	*Configuring EIGRP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Capabilities Manager

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3 Feature Information for Capabilities Manager

Feature Name	Releases	Feature Information
Capabilities Manager	15.1(1)SG 15.1(1)SY 15.2(2)S 15.2(3)T Cisco IOS XE Release 3.6S Cisco IOS XE Release 3.3SG	Capabilities Manager is enabled by default at system startup. At startup, it registers as a Service Routing Client and proceeds to discover various capabilities of the hardware and software platform. The following commands were introduced or modified: • **clear service-routing capabilities- manager** • **service-routing capabilities-manager** • **show service-routing plugins capman** • **show service-routing capabilities-manager internal** • **show service-routing capabilities-manager group**