# Configuring IP Routing Protocol-Independent Features

This module describes how to configure IP routing protocol-independent features. Some of the features discussed in this module include the Default Passive Interface, Fast-Switched Policy Routing, and Policy-Based Routing.



# Information About Basic IP Routing

## Variable-Length Subnet Masks

Dynamic routing protocols, such as the Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) Version 2, and static routes support variable-length subnet masks (VLSMs). VLSM enables an organization to use more than one subnet mask within the same network address space. VLSM allows you to conserve IP addresses and efficiently use the available address space. Implementing VLSM is often referred to as “subnetting a subnet.”

**Note** You may want to carefully consider the use of VLSMs. It is easy to make mistakes during address assignments and difficult to monitor networks that use VLSMs. The best way to implement VLSMs is to keep your existing addressing plan in place and gradually migrate some networks to VLSMs to recover address space.

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 256 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 subnet mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0–72.16.255.252) with 2 host addresses on each subnet.

**Note** To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Serial 0/0
Router(config-if)# ip address 172.16.255.5 255.255.255.252
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.16.0.0
```

# Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot build a route to a particular destination. They are also useful in specifying a gateway of last resort to which all unroutable packets will be sent.

To configure a static route, use the **ip route** command in global configuration mode.

Static routes remain in the router configuration until you remove them (by using the **no** form of the **ip route** command). However, you can override static routes with dynamic routing information through the assignment of administrative distance values. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers.

Each dynamic routing protocol has a default administrative distance, as listed in the table below. For a configured static route to be overridden, the administrative distance of the static route should be higher than that of the dynamic routing protocol.

***Table 1: Default Administrative Distances***

| Route Source | Default Administrative Distance |
|---|---|
| Connected interface | 0 |
| Static route | 1 |
| EIGRP summary route | 5 |
| External Border Gateway Protocol (BGP) | 20 |
| Internal EIGRP | 90 |
| Interior Gateway Routing Protocol (IGRP) | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| Exterior Gateway Protocol (EGP) | 140 |
| On-Demand Routing (ODR) | 160 |

| Route Source | Default Administrative Distance |
|---|---|
| External EIGRP | 170 |
| Internal BGP | 200 |
| Unknown | 255 |

Static routes that point to an interface are advertised through dynamic routing protocols, regardless of whether **redistribute static** router configuration commands were specified for those routing protocols. Static routes that point to an interface are advertised because the routing table considers these routes as connected routes and hence, these routes lose their static nature. However, if you define a static route to an interface that is not connected to one of the networks defined by a **network** command, no dynamic routing protocol will advertise the route unless a **redistribute static** command is specified for the protocols.

When an interface goes down, all static routes associated with that interface are removed from the IP routing table. Also, when the software can no longer find a valid next hop for the address specified as the address of the forwarding router in a static route, the static route is removed from the IP routing table.

# Default Routes

Default routes, also known as gateways of last resort, are used to route packets that are addressed to networks not explicitly listed in the routing table. A device might not be able to determine routes to all networks. To provide complete routing capability, network administrators use some devices as smart devices and give the remaining devices default routes to the smart device. (Smart devices have routing table information for the entire internetwork.) Default routes can be either passed along dynamically or configured manually into individual devices.

Most dynamic interior routing protocols include a mechanism for causing a smart device to generate dynamic default information, which is then passed along to other devices.

You can configure a default route by using the following commands:

- **ip default-gateway**
- **ip default-network**
- **ip route 0.0.0.0 0.0.0.0**

You can use the **ip default-gateway** global configuration command to define a default gateway when IP routing is disabled on a device. For instance, if a device is a host, you can use this command to define a default gateway for the device. You can also use this command to transfer a Cisco software image to a device when the device is in boot mode. In boot mode, IP routing is not enabled on the device.

Unlike the **ip default-gateway** command, the **ip default-network** command can be used when IP routing is enabled on a device. When you specify a network by using the **ip default-network** command, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateways of last resort configured by using the **ip default-network** command are propagated differently depending on which routing protocol is propagating the default route. For Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) to propagate the default route, the network specified by the **ip default-network** command must be known to IGRP or EIGRP. The network must be an IGRP- or EIGRP-derived network in the routing table, or the static route used to generate the route to the network must be redistributed into IGRP or EIGRP or advertised into these protocols by using the **network** command. The Routing Information Protocol (RIP) advertises a route to network 0.0.0.0 if a gateway of last

resort is configured by using the **ip default-network** command. The network specified in the **ip default-network** command need not be explicitly advertised under RIP.

Creating a static route to network 0.0.0.0 0.0.0.0 by using the **ip route 0.0.0.0 0.0.0.0** command is another way to set the gateway of last resort on a device. As with the **ip default-network** command, using the static route to 0.0.0.0 is not dependent on any routing protocols. However, IP routing must be enabled on the device. IGRP does not recognize a route to network 0.0.0.0. Therefore, it cannot propagate default routes created by using the **ip route 0.0.0.0 0.0.0.0** command. Use the **ip default-network** command to have IGRP propagate a default route.

EIGRP propagates a route to network 0.0.0.0, but the static route must be redistributed into the routing protocol.

Depending on your release of the Cisco software, the default route created by using the **ip route 0.0.0.0 0.0.0.0** command is automatically advertised by RIP devices. In some releases, RIP does not advertise the default route if the route is not learned via RIP. You might have to redistribute the route into RIP by using the **redistribute** command.

Default routes created using the **ip route 0.0.0.0 0.0.0.0** command are not propagated by Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Additionally, these default routes cannot be redistributed into OSPF or IS-IS by using the **redistribute** command. Use the **default-information originate** command to generate a default route into an OSPF or IS-IS routing domain.

## Default Network

Default networks are used to route packets to destinations not established in the routing table. You can use the **ip default-network** *network-number* global configuration command to configure a default network when IP routing is enabled on the device. When you configure a default network, the device considers routes to that network for installation as the gateway of last resort on the device.

## Gateway of Last Resort

When default information is being passed along through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the optimal default network as its default route. In the case of RIP, there is only one choice, network 0.0.0.0. In the case of EIGRP, there might be several networks that can be candidates for the system default. Cisco IOS software uses both the administrative distance and metric information to determine the default route (gateway of last resort). The selected default route appears in the gateway of last resort display of the **show ip route** command.

If dynamic default information is not being passed to the software, candidates for the default route are specified with the **ip default-network** global configuration command. In this usage, the **ip default-network** command takes an unconnected network as an argument. If this network appears in the routing table from any source (dynamic or static), the network is flagged as a candidate default route and is a possible choice as the default route.

If the router has no interface on the default network, but does have a route to the default network, the router considers this network as a candidate default path. The route candidates are examined and the best one is chosen based on the administrative distance and metric information. The gateway to the best default path becomes the gateway of last resort.

# Maximum Number of Paths

By default, most IP routing protocols install a maximum of four parallel paths in a routing table. Static routes always install six paths. The exception is BGP, which by default allows only one path (the best path) to the destination. However, BGP can be configured to use equal and unequal cost multipath load sharing. See the

"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" feature in the *BGP Configuration Guide* for more information.

The number of parallel paths that you can configure to be installed in the routing table is dependent on the installed version of the Cisco IOS software. To change the maximum number of parallel paths allowed, use the **maximum-paths** command in router configuration mode.

# Multi-Interface Load Splitting

Multi-interface load splitting allows you to efficiently control traffic that travels across multiple interfaces to the same destination. The **traffic-share min** router configuration command specifies that if multiple paths are available to the same destination, only paths with the minimum metric will be installed in the routing table. The number of paths allowed is never more than six. For dynamic routing protocols, the number of paths is controlled by the **maximum-paths** router configuration command. The static route source can install six paths. If more paths are available, the extra paths are discarded. If some installed paths are removed from the routing table, pending routes are added automatically.

# Routing Information Redistribution

You can configure the Cisco IOS software to redistribute information from one routing protocol to another. For example, you can configure a device to readvertise EIGRP-derived routes using RIP or to readvertise static routes using EIGRP. Redistribution from one routing protocol to another can be configured in all IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by configuring route maps between two domains. A route map is a route filter that is configured with permit and deny statements, match and set clauses, and sequence numbers. To define a route map for redistribution, use the **route-map** command in global configuration mode.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is hop count and the EIGRP metric is a combination of five metric values. In such situations, a dynamic metric is assigned to the redistributed route. Redistribution in these cases should be applied consistently and carefully in conjunction with inbound filtering to avoid the creation of routing loops.

The following examples illustrate the use of redistribution with and without route maps. The following example shows how to redistribute all OSPF routes into EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 101
Router(config-router)# exit
```

The following example shows how to redistribute RIP routes, with a hop count equal to 1, into OSPF. These routes will be redistributed into OSPF as external LSAs with a metric of 5, metric-type of type 1, and a tag equal to 1.

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

The following example shows how to redistribute OSPF learned routes with tag 7 as a RIP metric of 15:

```
Router(config)# router rip
Router(config-router)# redistribute ospf 1 route-map 5
Router(config-router)# exit
Router(config)# route-map 5 permit
Router(config-route-map)# match tag 7
Router(config-route-map)# set metric 15
```

The following example shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 into BGP with a metric of 5:

```
Router(config)# router bgp 50000
Router(config-router)# redistribute ospf 1 route-map 10
Router(config-router)# exit
Router(config)# route-map 10 permit
Router(config-route-map)# match route-type internal
Router(config-route-map)# match interface serial 0
Router(config-route-map)# set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP-derived CLNS prefix routes that match CLNS access list 2000; these routes are redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
Router(config)# router isis
Router(config-router)# redistribute ospf 1 route-map 2
Router(config-router)# redistribute iso-igrp nsfnet route-map 3
Router(config-router)# exit
Router(config)# route-map 2 permit
Router(config-route-map)# match route-type external
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# route-map 3 permit
Router(config-route-map)# match address 2000
Router(config-route-map)# set metric 30
Router(config-route-map)# exit
```

In the following example, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
Router(config)# router rip
Router(config-router)# redistribute ospf 101 route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 1 2
Router(config-route-map)# set metric 1
Router(config-route-map)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 3
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
Router(config)# route-map 1 deny
Router(config-route-map)# match tag 4
Router(config-route-map)# exit
Router(config)# route map 1 permit
```

```
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
```

The following example shows how a route map is referenced by using the **default-information** router configuration command. Such referencing is called conditional default origination. OSPF will generate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.16.0.0 is in the routing table.

```
Router(config)# route-map ospf-default permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type-2
Router(config-route-map)# exit
Router(config)# access-list 1 172.16.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

## Supported Automatic Metric Translations

This section describes supported automatic metric translations between routing protocols. The following points are based on the assumption that you have not defined a default redistribution metric that replaces metric conversions:

- RIP can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- BGP does not send metrics in its routing updates.
- EIGRP can automatically redistribute static routes from other EIGRP-routed autonomous systems as long as the static route and any associated interfaces are covered by an EIGRP network statement. EIGRP assigns static routes a metric that identifies them as directly connected. EIGRP does not change the metrics of routes derived from EIGRP updates from other autonomous systems.

**Note** Any protocol can redistribute routes from other routing protocols as long as a default metric is configured.

## Protocol Differences in Implementing the no redistribute Command

**Caution** Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting. In most cases, changing or disabling any keyword will not affect the state of other keywords.

Different protocols implement the **no redistribute** command differently as follows:

- In Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.

- The **no redistribute isis** command removes the Intermediate System to Intermediate System (IS-IS) redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- The Enhanced Interior Gateway Routing Protocol (EIGRP) used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.

# Default Passive Interfaces

The Default Passive Interfaces feature simplifies the configuration of distribution devices by allowing all interfaces to be set as passive by default. In ISPs and large enterprise networks, many distribution devices have more than 200 interfaces. Obtaining routing information from these interfaces requires configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on interfaces where adjacencies were not desired.

# Sources of Routing Information Filtering

Filtering sources of routing information prioritizes routing information gathered from different sources because some pieces of routing information may be more accurate than others. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value the lower the trust rating. An administrative distance of 255 means that the routing information source cannot be trusted at all and should be ignored.

In a large network, some routing protocols and some routers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running on the same router for IP, the same route may be advertised by more than one routing process. By specifying administrative distance values, you enable a router to intelligently discriminate between sources of routing information. The router will always pick the route whose routing protocol has the lowest administrative distance.

There are no guidelines for assigning administrative distances because each network has its own requirements. You must determine a reasonable matrix of administrative distances for a network as a whole.

**Note** You can use the administrative distance to rate the routing information from routers that are running the same routing protocol. However, using the administrative distance for this purpose can result in inconsistent routing information and forwarding loops.

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command specifies EIGRP routing on networks 192.0.2.16 and 172.16.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the router to ignore all routing updates from routers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
Router(config)# router eigrp 1
Router(config-router)# network 192.0.2.16
Router(config-router)# network 172.16.0.0
```

```
Router(config-router)# distance 255
Router(config-router)# distance eigrp 80 100
Router(config-router)# distance 120 172.16.1.3 0.0.0.0
```

**Note** The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

The following example assigns the router with the address 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
```

**Note** Administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the administrative distance value for learned IP routes is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
Router(config)# router isis
Router(config-router)# distance 90 ip
```

# Policy-Based Routing

Policy-based routing (PBR) is a more flexible mechanism than destination routing for routing packets. It is a process whereby a router puts packets through a route map before routing them. The route map determines which packets are routed to which router next. You can enable PBR if you want certain packets to be routed some way other than the obvious shortest path. Possible applications for policy-based routing include protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, and routing based on dedicated links.

To enable PBR, you must identify the route map to be used for PBR and create the route map. The route map specifies the match criteria and the resulting action if all match clauses are met.

A packet arriving on a specified interface will be subject to PBR, except when its destination IP address is the same as the IP address of the router's interface. To disable fast switching of all packets arriving on this interface, use the **ip policy route-map** command in interface configuration mode.

To define the route map to be used for PBR, use the **route-map** command in global configuration mode.

To define the criteria by which packets are examined to learn if they will follow PBR, use either the **match length** command or the **match ip address** command or both in route map configuration mode. The **match length** command allows you to configure policy routing based on the Level 3 length of the packet, and the

**match ip address** command allows you to policy route packets based on the criteria that can be matched with an extended access list.

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1 from the source 10.1.1.1 are sent to the router at 172.16.6.6 if the router has no explicit route for the destination of the packets. Packets that arrive from the source 172.17.2.2 are sent to the router at 192.168.7.7 if the router has no explicit route for the destination of the packets. All other packets for which the router has no explicit route to the destination are discarded.

```
Router(config)# access-list 1 permit ip 10.1.1.1
Router(config)# access-list 2 permit ip 172.17.2.2
Router(config)# interface async 1
Router(config-if)# ip policy route-map equal-access
Router(config-if)# exit
Router(config)# route-map equal-access permit 10
Router(config-route-map)# match ip address 1
Router(config-route-map)# set ip default next-hop 172.16.6.6
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 20
Router(config-route-map)# match ip address 2
Router(config-route-map)# set ip default next-hop 192.168.7.7
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 30
Router(config-route-map)# set default interface null 0
Router(config-route-map)# exit
```

You can set IP header precedence bits in the router when PBR is enabled. The precedence setting in the IP header determines how packets are treated during times of high traffic. When packets containing these headers arrive at another router, the packets are ordered for transmission according to the precedence set if the queuing feature is enabled. The router does not honor the precedence bits if queuing is not enabled, and the packets are sent in FIFO order. You can change the precedence setting by using either a number or a name.

The table below lists the possible IP Precedence values (numbers and their corresponding names), from the least important to the most important.

***Table 2: IP Precedence Values***

| Number | Name |
|---|---|
| 0 | routine |
| 1 | priority |
| 2 | immediate |
| 3 | flash |
| 4 | flash-override |
| 5 | critical |
| 6 | internet |
| 7 | network |

## Fast-Switched Policy Routing

IP policy routing can be fast-switched. Prior to fast-switched policy routing, policy routing could only be process -switched, which meant that on most platforms, the switching rate was approximately 1000 to 10,000 packets per second. Such rates were not fast enough for many applications. With fast-switched policy routing, users who need policy routing to occur at faster speeds can implement policy routing without slowing down the device.

Fast-switched policy routing supports all **match** commands and most **set** commands, except for the following:

- **set ip default**
- **set interface**

The **set interface** command is supported only over point-to-point links, unless there is a route cache entry that uses the same interface that is specified in the command in the route map.

To configure fast-switched policy routing, use the **ip route-cache policy** interface configuration command.

## Local Policy Routing

Packets that are generated by the router are not normally policy-routed. To enable local policy routing for such packets, you must indicate which route map the router should use. All packets originating on the router will then be subject to local policy routing. To identify the route map to be used for local policy routing, use the **ip local policy route-map** command in global configuration mode.

Use the **show ip local policy** command to display the route map used for local policy routing, if one exists.

## NetFlow Policy Routing

NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and information monitoring on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (formerly known as CEF), distributed Cisco Express Forwarding (formerly known as dCEF), and NetFlow.

NetFlow policy routing leverages the following technologies:

- Cisco Express Forwarding, which looks at a Forwarding Information Base (FIB) instead of a routing table when switching packets, to address maintenance problems of a demand caching scheme.
- Distributed Cisco Express Forwarding, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides accounting, capacity planning, and traffic monitoring capabilities.

The following are the benefits of NPR:

- NPR takes advantage of new switching services. Cisco Express Forwarding, distributed Cisco Express Forwarding, and NetFlow can now use policy routing.
- Policy routing can be deployed on a wide scale and on high-speed interfaces.

NPR is the default policy routing mode. No additional configuration tasks are required to enable policy routing with Cisco Express Forwarding, distributed Cisco Express Forwarding, or NetFlow. As soon as one of these features is turned on, packets are automatically subjected to policy routing in the appropriate switching path.

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip precedence priority
Device(config-route-map)# set ip next-hop 10.0.0.8
Device(config-route-map)# set ip next-hop verify-availability
Device(config-route-map)# exit
Device(config)# route-map test permit 20
Device(config-route-map)# match ip address 101
Device(config-route-map)# set interface Ethernet 0/0/3
Device(config-route-map)# set ip tos max-throughput
Device(config-route-map)# exit
```

## Authentication Key Management and Supported Protocols

Key management is a method of controlling the authentication keys used by routing protocols. Not all protocols support key management. Authentication keys are available for Director Response Protocol (DRP) Agent, Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP) Version 2.

You can manage authentication keys by defining key chains, identifying the keys that belong to the key chain, and specifying how long each key is valid. Each key has its own key identifier (specified using the **key chain** configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the message digest algorithm 5 (MD5) authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes.

# How to Configure Basic IP Routing

## Redistributing Routing Information

You can redistribute routes from one routing domain into another, with or without controlling the redistribution with a route map. To control which routes are redistributed, configure a route map and reference the route map from the **redistribute** command.

The tasks in this section describe how to define the conditions for redistributing routes (a route map), how to redistribute routes, and how to remove options for redistributing routes, depending on the protocol being used.

## Defining Conditions for Redistributing Routes

Route maps can be used to control route redistribution (or to implement policy-based routing). To define conditions for redistributing routes from one routing protocol into another, configure the **route-map** command. Then use at least one **match** command in route map configuration mode, as needed. At least one **match** command is used in this task because the purpose of the task is to illustrate how to define one or more conditions on which to base redistribution.

**Note** A route map is not required to have **match** commands; it can have only **set** commands. If there are no **match** commands, everything matches the route map.

**Note** There are many more **match** commands not shown in this table. For additional **match** commands, see the *Cisco IOS Master Command List*.

| Command or Action | Purpose |
|---|---|
| **match as-path** *path-list-number* | Matches a BGP autonomous system path access list. |
| **match community** {*standard-list-number* \| *expanded-list-number* \| *community-list-name* **match community**[**exact**]} | Matches a BGP community. |
| **match ip address** {*access-list-number* [*access-list-number...* \| *access-list-name...*] \| *access-list-name* [*access-list-number...*\| *access-list-name*] \| **prefix-list** *prefix-list-name* [*prefix-list-name...*]} | Matches routes that have a destination network address that is permitted to policy route packets or is permitted by a standard access list, an extended access list, or a prefix list. |
| **match metric** *metric-value* | Matches routes with the specified metric. |
| **match ip next-hop** {*access-list-number* \| *access-list-name*} [*access-list-number* \| *access-list-name*] | Matches a next-hop device address passed by one of the specified access lists. |
| **match tag** *tag-value* [*tag-value*] | Matches the specified tag value. |
| **match interface** *type number* [*type number*] | Matches routes that use the specified interface as the next hop. |
| **match ip route-source** {*access-list-number* \| *access-list-name*} [*access-list-number* \| *access-list-name*] | Matches the address specified by the advertised access lists. |

| Command or Action | Purpose |
|---|---|
| **match route-type** {**local** \| **internal** \| **external** [**type-1** \| **type-2**] \| **level-1** \| **level-2**} | Matches the specified route type. |

To optionally specify the routing actions for the system to perform if the match criteria are met (for routes that are being redistributed by the route map), use one or more **set** commands in route map configuration mode, as needed.

**Note** A route map is not required to have **set** commands; it can have only **match** commands.

**Note** There are more **set** commands not shown in this table. For additional **set** commands, see the *Cisco IOS Master Command List*.

| Command or Action | Purpose |
|---|---|
| **set community** {*community-number* [**additive**] [**well-known**]\| **none**} | Sets the community attribute (for BGP). |
| **set dampening** *halflife reuse suppress max-suppress-time* | Sets route dampening parameters (for BGP). |
| **set local-preference** *number-value* | Assigns a local preference value to a path (for BGP). |
| **set origin** {**igp** \| **egp** *as-number* \| **incomplete**} | Sets the route origin code. |
| **set as-path**{**tag** \| **prepend** *as-path-string* } | Modifies the autonomous system path (for BGP). |
| **set next-hop** *next-hop* | Specifies the address of the next hop. |
| **set automatic-tag** | Enables automatic computation of the tag table. |
| **set level** {**level-1** \| **level-2** \| **level-1-2** \| **stub-area** \| **backbone**} | Specifies the areas to import routes. |
| **set metric** *metric-value* | Sets the metric value for redistributed routes (for any protocol, except EIGRP). |

| Command or Action | Purpose |
|---|---|
| **set metric** *bandwidth delay reliability load mtu* | Sets the metric value for redistributed routes (for EIGRP only). |
| **set metric-type** {**internal** \| **external** \| **type-1** \| **type-2**} | Sets the metric type for redistributed routes. |
| **set metric-type internal** | Sets the Multi Exit Discriminator (MED) value on prefixes advertised to the external BGP neighbor to match the Interior Gateway Protocol (IGP) metric of the next hop. |
| **set tag** *tag-value* | Sets a tag value to be applied to redistributed routes. |

## Redistributing Routes from One Routing Domain to Another

Perform this task to redistribute routes from one routing domain into another and to control route redistribution. This task shows how to redistribute OSPF routes into a BGP domain.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system*
4. **redistribute** *protocol process-id*
5. **default-metric** *number*
6. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **router bgp** *autonomous-system*<br>**Example:**<br>`Device(config)# router bgp 109` | Enables a BGP routing process and enters router configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **redistribute** *protocol process-id*<br>**Example:**<br>`Device(config-router)# redistribute ospf 2 1` | Redistributes routes from the specified routing domain into another routing domain. |
| **Step 5** | **default-metric** *number*<br>**Example:**<br>`Device(config-router)# default-metric 10` | Sets the default metric value for redistributed routes.<br>**Note** The metric value specified in the **redistribute** command supersedes the metric value specified using the **default-metric** command. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-router)# end` | Exits router configuration mode and returns to privileged EXEC mode. |

## Removing Options for Redistribution Routes

**Caution** Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting.

Different protocols implement the **no redistribute** command differently as follows:

- In BGP, OSPF, and RIP configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the IS-IS redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- EIGRP used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.
- For the **no redistribute connected** command, the behavior is subtractive if the **redistribute** command is configured under the **router bgp** or the **router ospf** command. The behavior is complete removal of the command if it is configured under the **router isis** or the **router eigrp** command.

The following OSPF commands illustrate how various options are removed from the redistribution in router configuration mode.

| Command or Action | Purpose |
|---|---|
| `no redistribute connected metric 1000 subnets` | Removes the configured metric value of 1000 and the configured subnets and retains the **redistribute connected** command in the configuration. |
| `no redistribute connected metric 1000` | Removes the configured metric value of 1000 and retains the **redistribute connected subnets** command in the configuration. |
| `no redistribute connected subnets` | Removes the configured subnets and retains the **redistribute connected metric** *metric-value* command in the configuration. |
| `no redistribute connected` | Removes the **redistribute connected** command and any of the options that were configured for the command. |

# Configuring Routing Information Filtering

To filter routing protocol information, perform the tasks in this section.

**Note** When routes are redistributed between OSPF processes, no OSPF metric is preserved.

## Preventing Routing Updates Through an Interface

To prevent other routers on a local network from dynamically learning routes, you can keep routing update messages from being sent through a router interface. To prevent routing updates through a specified interface, use the **passive-interface** command in router configuration mode. This command is supported in all IP-based routing protocols, except BGP.

OSPF and IS-IS behave differently. In OSPF, the interface address that you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified router interface. In IS-IS, the specified IP addresses are advertised without actually running IS-IS on those interfaces.

## Configuring Default Passive Interfaces

Perform this task to set all interfaces on a device, in an Enhanced Interior Gateway Routing Protocol (EIGRP) environment, as passive by default, and then activate only those interfaces where adjacencies are desired.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** {*autonomous-system-number* | *virtual-instance-number*}
4. **passive-interface** [**default**] [*type number*]
5. **no passive-interface** [**default**] [*type number*]
6. **network** *network-address* [*options*]
7. **end**
8. **show ip eigrp interfaces**
9. **show ip interface**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **router eigrp** {*autonomous-system-number* \| *virtual-instance-number*}<br>**Example:**<br>`Device(config)# router eigrp 1` | Configures an EIGRP process and enters router configuration mode.<br>• *autonomous-system-number*—Autonomous system number that identifies the services to the other EIGRP address-family devices. It is also used to tag routing information. The range is 1 to 65535.<br>• *virtual-instance-number*—EIGRP virtual instance name. This name must be unique among all address-family router processes on a single device, but need not be unique among devices |
| **Step 4** | **passive-interface** [**default**] [*type number*]<br>**Example:**<br>`Device(config-router)# passive-interface default` | Sets all interfaces as passive by default. |
| **Step 5** | **no passive-interface** [**default**] [*type number*]<br>**Example:**<br>`Device(config-router)# no passive-interface gigabitethernet 0/0/0` | Activates only those interfaces that need adjacencies. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **network** *network-address* [*options*]<br>**Example:**<br>`Device(config-router)# network 192.0.2.0` | Specifies the list of networks to be advertised by routing protocols. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config-router)# end` | Exits router configuration mode and returns to privileged EXEC mode. |
| **Step 8** | **show ip eigrp interfaces**<br>**Example:**<br>`Device# show ip eigrp interfaces` | Verifies whether interfaces on your network have been set to passive. |
| **Step 9** | **show ip interface**<br>**Example:**<br>`Device# show ip interface` | Verifies whether interfaces you enabled are active. |

## Controlling the Advertising of Routes in Routing Updates

To prevent other devices from learning one or more routes, you can suppress routes from being advertised in routing updates. To suppress routes from being advertised in routing updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing-process* | *as-number*] command in router configuration mode.

You cannot specify an interface name in Open Shortest Path First (OSPF). When used for OSPF, this feature applies only to external routes.

## Controlling the Processing of Routing Updates

You might want to avoid processing certain routes listed in incoming updates. This feature does not apply to OSPF or IS-IS. To suppress routes in incoming updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **in** [*interface-type interface-number*] command in router configuration mode.

## Filtering Sources of Routing Information

To filter sources of routing information, use the **distance** *ip-address wildcard- mask* [*ip-standard-acl* | *ip-extended-acl* | *access-list-name*] command in router configuration mode.

# Configuring Precedence for Policy-Based Routing Default Next-Hop Routes

Perform this task to configure the precedence of packets and specify where packets that pass the match criteria are output.

**Note** The **set ip next-hop** and **set ip default next-hop** commands are similar but have a different order of operation. Configuring the **set ip next-hop** command causes the system to first use policy routing and then use the routing table. Configuring the **set ip default next-hop** command causes the system to first use the routing table and then the policy-route-specified next hop.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **set ip precedence** {*number* | *name*}
5. **set ip next-hop** *ip-address* [*ip-address*]
6. **set interface** *type number* [*...type number*]
7. **set ip default next-hop** *ip-address* [*ip-address*]
8. **set default interface** *type number* [*...type number*]
9. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **route-map** *map-tag* [**permit** \| **deny**] [*sequence-number*] [<br>**Example:**<br>`Device(config)# route-map alpha permit ordering-seq` | Configures a route map and specifies how the packets are to be distributed. |
| **Step 4** | **set ip precedence** {*number* \| *name*}<br>**Example:**<br>`Device(config-route-map)# set ip precedence 5` | Sets the precedence value in the IP header.<br>**Note** You can specify either a precedence number or a precedence name. |
| **Step 5** | **set ip next-hop** *ip-address* [*ip-address*]<br>**Example:**<br>`Device(config-route-map)# set ip next-hop 192.0.2.1` | Specifies the next hop for routing packets.<br>**Note** The next hop must be an adjacent device. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **set interface** *type number* [*...type number*]<br>**Example:**<br>`Device(config-route-map)# set interface gigabitethernet 0/0/0` | Specifies the output interface for the packet. |
| **Step 7** | **set ip default next-hop** *ip-address* [*ip-address*]<br>**Example:**<br>`Device(config-route-map)# set ip default next-hop 172.16.6.6` | Specifies the next hop for routing packets if there is no explicit route for this destination.<br>**Note** Like the **set ip next-hop** command, the **set ip default next-hop** command must specify an adjacent device. |
| **Step 8** | **set default interface** *type number* [*...type number*]<br>**Example:**<br>`Device(config-route-map)# set default interface serial 0/0/0` | Specifies the output interface for the packet if there is no explicit route for the destination. |
| **Step 9** | **end**<br>**Example:**<br>`Device(config-route-map)# end` | Exits route-map configuration mode and returns to privileged EXEC mode. |

# Configuring QoS Policy Propagation via BGP

## Configuring QoS Policy Propagation via BGP Based on Community Lists

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*]
11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **exit**
14. **ip bgp-community new-format**
15. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **route-map** *map-tag* [**permit** \| **deny**] [*sequence-number*] [<br>**Example:**<br>`Device(config)# route-map alpha permit ordering-seq` | Configures a route map and specifies how the packets are to be distributed. . |
| **Step 4** | **match community** {*standard-list-number* \| *expanded-list-number* \| *community-list-name* [**exact**]}<br>**Example:**<br>`Device(config-route-map)# match community 1` | Matches a Border Gateway Protocol (BGP) community list. |
| **Step 5** | **set ip precedence** [*number* \| *name*]<br>**Example:**<br>`Device(config-route-map)# set ip precedence 5` | Sets the IP Precedence field when the community list matches.<br>**Note** You can specify either a precedence number or a precedence name. |
| **Step 6** | **exit**<br>**Example:**<br>`Device(config-route-map)# exit` | Exits route-map configuration mode and returns to global configuration mode. |
| **Step 7** | **router bgp** *autonomous-system*<br>**Example:**<br>`Device(config)# router bgp 45000` | Enables a BGP process and enters router configuration mode. |
| **Step 8** | **table-map** *route-map-name*<br>**Example:**<br>`Device(config-router)# table-map rm1` | Modifies the metric and tag values when the IP routing table is updated with BGP learned routes. |
| **Step 9** | **exit**<br>**Example:** | Exits router configuration mode and returns to global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config-router)# exit` | |
| **Step 10** | **ip community-list** *standard-list-number* {**permit** \| **deny**} [*community-number*]<br>**Example:**<br>`Device(config)# ip community-list 1 permit 2` | Creates a community list for BGP and controls access to it. |
| **Step 11** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface gigabitethernet 0/0/0` | Specifies the interface (or subinterface) and enters interface configuration mode. |
| **Step 12** | **bgp-policy** {**source** \| **destination**} **ip-prec-map**<br>**Example:**<br>`Device(config-if)# bgp-policy source ip-prec-map` | Classifies packets using IP precedence. |
| **Step 13** | **exit**<br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode and returns to global configuration mode. |
| **Step 14** | **ip bgp-community new-format**<br>**Example:**<br>`Device(config)# ip bgp-community new-format` | (Optional) Displays the BGP community number in AA:NN (autonomous system:community number/4-byte number) format. |
| **Step 15** | **end**<br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and returns to privileged EXEC mode. |

## Configuring QoS Policy Propagation via BGP Based on the Autonomous System Path Attribute

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable** ]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [ **ordering-seq** *sequence-name*
5. **match as-path** *path-list-number*
6. **set ip precedence** [*number* | *name*]
7. **exit**
8. **router bgp** *autonomous-system*
9. **table-map** *route-map-name*

10. **exit**
11. **ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression*
12. **interface** *type number*
13. **bgp-policy** {**source** | **destination**} **ip-prec-map**
14. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **named-ordering-route-map enable** ]<br>**Example:**<br>`Device(config)# named-ordering-route-map enable` | Enables ordering of route-maps based on a string provided by the user. |
| **Step 4** | **route-map** *map-tag* [**permit** \| **deny**] [*sequence-number*] [ **ordering-seq** *sequence-name*<br>**Example:**<br>`Device(config)# route-map alpha permit ordering-seq sequence1` | Configures a route map and specifies how the packets are to be distributed. **ordering-seq** indicates the sequence that is to be used for ordering of route-maps. |
| **Step 5** | **match as-path** *path-list-number*<br>**Example:**<br>`Device(config-route-map)# match as-path 2` | Matches a Border Gateway Protocol (BGP) autonomous system path access list. |
| **Step 6** | **set ip precedence** [*number* \| *name*]<br>**Example:**<br>`Device(config-route-map)# set ip precedence 5` | Sets the IP Precedence field when the autonomous-system path matches.<br>**Note** You can specify either a precedence number or a precedence name. |
| **Step 7** | **exit**<br>**Example:**<br>`Device(config-route-map)# exit` | Exits route-map configuration mode and returns to global configuration mode. |
| **Step 8** | **router bgp** *autonomous-system*<br>**Example:**<br>`Device(config)# router bgp 45000` | Enables a BGP process and enters router configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **table-map** *route-map-name*<br>**Example:**<br>`Device(config-router)# table-map rm1` | Modifies the metric and tag values when the IP routing table is updated with BGP learned routes. |
| **Step 10** | **exit**<br>**Example:**<br>`Device(config-router)# exit` | Exits router configuration mode and returns to global configuration mode. |
| **Step 11** | **ip as-path access-list** *access-list-number* {**permit** \| **deny**} *as-regular-expression*<br>**Example:**<br>`Device(config)# ip as-path access-list 500 permit 45000` | Defines an autonomous system path access list. |
| **Step 12** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface gigabitethernet 0/0/0` | Specifies the interface (or subinterface) and enters interface configuration mode. |
| **Step 13** | **bgp-policy** {**source** \| **destination**} **ip-prec-map**<br>**Example:**<br>`Device(config-if)# bgp-policy source ip-prec-map` | Classifies packets using IP precedence. |
| **Step 14** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

## Configuring QoS Policy Propagation Based on an Access List

This section describes how to configure the QoS Policy Propagation via BGP feature based on an access list. This section assumes that you have already configured Cisco Express Forwarding or distributed Cisco Express Forwarding and BGP on your router.

Perform this task to configure the router to propagate the IP precedence based on an access list:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]]
4. **match ip address** *access-list-number*
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **access-list** *access-list-number* {**permit** | **deny**} *source*

11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **route-map** *route-map-name* [**permit** \| **deny** [*sequence-number*]]<br>**Example:**<br>`Router(config)# route-map rm1` | Defines a route map to control redistribution and enters route-map configuration mode. |
| **Step 4** | **match ip address** *access-list-number*<br>**Example:**<br>`Router(config-route-map)# match ip address 3` | Matches routes that have a destination network address that is permitted by a standard or extended access list. |
| **Step 5** | **set ip precedence** [*number* \| *name*]<br>**Example:**<br>`Router(config-route-map)# set ip precedence 5` | Sets the IP Precedence field when the autonomous system path matches.<br>**Note** You can specify either a precedence number or a precedence name. |
| **Step 6** | **exit**<br>**Example:**<br>`Router(config-route-map)# exit` | Exits route-map configuration mode and returns to global configuration mode. |
| **Step 7** | **router bgp** *autonomous-system*<br>**Example:**<br>`Router(config)# router bgp 45000` | Enables a BGP routing process and enters router configuration mode. |
| **Step 8** | **table-map** *route-map-name*<br>**Example:**<br>`Router(config-router)# table-map rm1` | Modifies the metric and tag values when the IP routing table is updated with BGP learned routes. |
| **Step 9** | **exit**<br>**Example:**<br>`Router(config-router)# exit` | Exits router configuration mode and returns to global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 10** | **access-list** *access-list-number* {**permit** \| **deny**} *source*<br>**Example:**<br>`Router(config)# access-list 2 permit 172.16.0.2` | Defines an access list. |
| **Step 11** | **interface** *type number*<br>**Example:**<br>`Router(config)# interface ethernet 0/0` | Specifies the interface (or subinterface) and enters interface configuration mode. |
| **Step 12** | **bgp-policy** {**source** \| **destination**} **ip-prec-map**<br>**Example:**<br>`Router(config-if)# bgp-policy source ip-prec-map` | Classifies packets using IP precedence. |
| **Step 13** | **end**<br>**Example:**<br>`Router(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

## Monitoring QoS Policy Propagation via BGP

To monitor the QoS Policy Propagation via the BGP feature configuration, use the following optional commands.

| Command or Action | Purpose |
|---|---|
| **show ip bgp** | Displays entries in the Border Gateway Protocol (BGP) routing table to verify whether the correct community is set on the prefixes. |
| **show ip bgp community-list** *community-list-number* | Displays routes permitted by the BGP community to verify whether correct prefixes are selected. |
| **show ip cef** *network* | Displays entries in the forwarding information base (FIB) table based on the specified IP address to verify whether Cisco Express Forwarding has the correct precedence value for the prefix. |
| **show ip interface** | Displays information about the interface. |

| Command or Action | Purpose |
|---|---|
| `show ip route` *prefix* | Displays the current status of the routing table to verify whether correct precedence values are set on the prefixes. |

# Managing Authentication Keys

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *number*
5. **key-string** *text*
6. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
7. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
8. **end**
9. **show key chain**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes.<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **key chain** *name-of-chain*<br>**Example:**<br>`Device(config)# key chain chain1` | Defines a key chain and enters key-chain configuration mode. |
| **Step 4** | **key** *number*<br>**Example:**<br>`Device(config-keychain)# key 1` | Identifies number of an authentication key on a key chain. The range of keys is from 0 to 2147483647. The key identification numbers need not be consecutive. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **key-string** *text*<br>**Example:**<br>`Device(config-keychain-key)# key-string string1` | Identifies the key string. |
| **Step 6** | **accept-lifetime** *start-time* {**infinite** \| *end-time* \| **duration** *seconds*}<br>**Example:**<br>`Device(config-keychain-key)# accept-lifetime 13:30:00 Dec 22 2011 duration 7200` | Specifies the time period during which the key can be received. |
| **Step 7** | **send-lifetime** *start-time* {**infinite** \| *end-time* \| **duration** *seconds*}<br>**Example:**<br>`Device(config-keychain-key)# send-lifetime 14:30:00 Dec 22 2011 duration 3600` | Specifies the time period during which the key can be sent. |
| **Step 8** | **end**<br>**Example:**<br>`Device(config-keychain-key)# end` | Exits key-chain key configuration mode and returns to privileged EXEC mode. |
| **Step 9** | **show key chain**<br>**Example:**<br>`Device# show key chain` | (Optional) Displays authentication key information. |

# Monitoring and Maintaining the IP Network

## Clearing Routes from the IP Routing Table

You can remove all contents of a particular table. Clearing a table may become necessary when the contents of the particular structure have become, or are suspected to be, invalid.

To clear one or more routes from the IP routing table, use the **clear ip route** {*network* [*mask*] | *****} command in privileged EXEC mode.

## Displaying System and Network Statistics

You can use the following **show** commands to display system and network statistics. You can display specific statistics such as contents of IP routing tables, caches, and databases. You can also display information about node reachability and discover the routing path that packets leaving your device are taking through the network. This information can an be used to determine resource utilization and solve network problems.

| Command or Action | Purpose |
|---|---|
| **`show ip cache policy`** | Displays cache entries in the policy route cache. |
| **`show ip local policy`** | Displays the local policy route map if one exists. |

| Command or Action | Purpose |
|---|---|
| **show ip policy** | Displays policy route maps. |
| **show ip protocols** | Displays the parameters and current state of the active routing protocols. |
| **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] \| *protocol* [*process-id*] \| **list** {**access-list-number** \| *access-list-name*} \| **static download**] | Displays the current state of the routing table. |
| **show ip route summary** | Displays the current state of the routing table in summary form. |
| **show ip route supernets-only** | Displays supernets. |
| **show key chain** [*name-of-chain*] | Displays authentication key information. |
| **show route-map** [*map-name*] | Displays all route maps configured or only the one specified. |

# Configuration Examples for Basic IP Routing

## Example: Variable-Length Subnet Mask

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 256 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0 - 172.16.255.252) with 2 host addresses on each subnet.

**Danger** To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

```
Router(config)# interface Ethernet 0/0

Router(config-if)# ip address 172.16.1.1 255.255.255.0

Router(config-if)# ! 8 bits of host address space reserved for Ethernet interfaces
Router(config-if)# exit
Router(config)# interface Serial 0/0

Router(config-if)# ip address 172.16.255.5 255.255.255.252

Router(config-if)# ! 2 bits of address space reserved for point-to-point serial interfaces
```

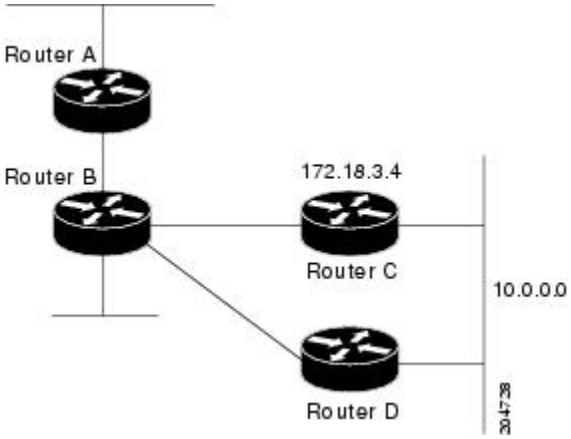
```
Router(config-if)# exit

Router(config)# router rip
Router(config-router)# network 172.16.0.0
Router(config-router)# ! Specifies the network directly connected to the router
```

# Example: Overriding Static Routes with Dynamic Protocols

In the following example, packets for network 10.0.0.0 from Router B (where the static route is installed) will be routed through 172.18.3.4 if a route with an administrative distance less than 110 is not available. The figure below illustrates this example. The route learned by a protocol with an administrative distance of less than 110 might cause Router B to send traffic destined for network 10.0.0.0 via the alternate path--through Router D.

```
Router(config)# ip route 10.0.0.0 255.0.0.0 172.18.3.4 110
```

***Figure 1: Overriding Static Routes***




# Example: Administrative Distances

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command specifies EIGRP routing on networks 192.0.2.16 and 172.16.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the router to ignore all routing updates from routers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
Router(config)# router eigrp 1
Router(config-router)# network 192.0.2.16
Router(config-router)# network 172.16.0.0
Router(config-router)# distance 255
Router(config-router)# distance eigrp 80 100
Router(config-router)# distance 120 172.16.1.3 0.0.0.0
```

**Note** The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

The following example assigns the router with the address 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
```

**Note** Assigning administrative distances can be used to solve unique problems. However, administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the distance value for learned IP routes is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
Router(config)# router isis

Router(config-router)# distance 90 ip
```

# Example: Static Routing Redistribution

In the example that follows, three static routes are specified, two of which are to be advertised. The static routes are created by specifying the **redistribute static** router configuration command and then specifying an access list that allows only those two networks to be passed to the EIGRP process. Any redistributed static routes should be sourced by a single router to minimize the likelihood of creating a routing loop.

```
Router(config)# ip route 192.168.2.0 255.255.255.0 192.168.7.65
Router(config)# ip route 192.168.5.0 255.255.255.0 192.168.7.65
Router(config)# ip route 10.10.10.0 255.255.255.0 10.20.1.2
Router(config)# !
Router(config)# access-list 3 permit 192.168.2.0 0.0.255.255
Router(config)# access-list 3 permit 192.168.5.0 0.0.255.255
Router(config)# access-list 3 permit 10.10.10.0 0.0.0.255
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.10.10.0
Router(config-router)# redistribute static metric 10000 100 255 1 1500
Router(config-router)# distribute-list 3 out static
```

# Example: EIGRP Redistribution

Each EIGRP routing process provides routing information to only one autonomous system. The Cisco IOS software must run a separate EIGRP process and maintain a separate routing database for each autonomous system that the software services. However, you can transfer routing information among routing databases.

In the following example, network 10.0.0.0 is configured under EIGRP autonomous system 1 and network 192.168.7.0 is configured under EIGRP autonomous system 101:

```
Router(config)# router eigrp 1
Router(config-router)# network 10.0.0.0
Router(config-router)# exit
Router(config)# router eigrp 101
Router(config-router)# network 192.168.7.0
```

In the following example, routes from the 192.168.7.0 network are redistributed into autonomous system 1 (without passing any other routing information from autonomous system 101):

```
Router(config)# access-list 3 permit 192.168.7.0
Router(config)# !
Router(config)# route-map 101-to-1 permit 10
Router(config-route-map)# match ip address 3
Router(config-route-map)# set metric 10000 100 1 255 1500
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101 route-map 101-to-1
Router(config-router)#!
```

The following example is an alternative way to redistribute routes from the 192.168.7.0 network into autonomous system 1.This method does not allow you to set the metric for redistributed routes.

```
Router(config)# access-list 3 permit 192.168.7.0
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101
Router(config-router)# distribute-list 3 out eigrp 101
Router(config-router)# !
```

# Example: Mutual Redistribution Between EIGRP and RIP

Consider a WAN at a university that uses the Routing Information Protocol (RIP) as an interior routing protocol. Assume that the university wants to connect its WAN to regional network 172.16.0.0, which uses the Enhanced Interior Gateway Routing Protocol (EIGRP) as the routing protocol. The goal in this case is to advertise the networks in the university network to devices in the regional network.

Mutual redistribution is configured between EIGRP and RIP in the following example:

```
Device(config)# access-list 10 permit 172.16.0.0
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip metric 10000 100 255 1 1500
Device(config-router)# default-metric 10
Device(config-router)# distribute-list 10 out rip
Device(config-router)# exit
Device(config)# router rip
```

```
Device(config-router)# redistribute eigrp 1
Device(config-router)# !
```

In this example, an EIGRP routing process is started. The **network** router configuration command specifies that network 172.16.0.0 (the regional network) is to send and receive EIGRP routing information. The **redistribute** router configuration command specifies that RIP-derived routing information be advertised in routing updates. The **default-metric** router configuration command assigns an EIGRP metric to all RIP-derived routes. The **distribute-list** router configuration command instructs the Cisco software to use access list 10 (not defined in this example) to limit the entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

# Example: Mutual Redistribution Between EIGRP and BGP

In the following example, mutual redistribution is configured between the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Border Gateway Protocol (BGP).

Routes from EIGRP routing process 101 are injected into BGP autonomous system 50000. A filter is configured to ensure that the correct routes are advertised, in this case, three networks. Routes from BGP autonomous system 50000 are injected into EIGRP routing process 101. The same filter is used.

```
Device(config)# ! All networks that should be advertised from R1 are controlled with ACLs:

Device(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.16.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.25.0.0 0.0.255.255
Device(config)# ! Configuration for router R1:
Device(config)# router bgp 50000
Device(config-router)# network 172.18.0.0
Device(config-router)# network 172.16.0.0
Device(config-router)# neighbor 192.168.10.1 remote-as 2
Device(config-router)# neighbor 192.168.10.15 remote-as 1
Device(config-router)# neighbor 192.168.10.24 remote-as 3
Device(config-router)# redistribute eigrp 101
Device(config-router)# distribute-list 1 out eigrp 101
Device(config-router)# exit
Device(config)# router eigrp 101
Device(config-router)# network 172.25.0.0
Device(config-router)# redistribute bgp 50000
Device(config-router)# distribute-list 1 out bgp 50000
Device(config-router)# !
```

**Caution** BGP should be redistributed into an Interior Gateway Protocol (IGP) when there are no other suitable options. Redistribution from BGP into any IGP should be applied with proper filtering by using distribute lists, IP prefix lists, and route map statements to limit the number of prefixes.

# Examples: OSPF Routing and Route Redistribution

OSPF typically requires coordination among many internal devices, area border routers (ABRs), and Autonomous System Boundary Routers (ASBRs). At a minimum, OSPF-based devices can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas.

This section provides the following configuration examples:

- The first example shows simple configurations illustrating basic OSPF commands.

- The second example shows configurations for an internal device, ABR, and ASBR within a single, arbitrarily assigned OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

## Example: Basic OSPF Configurations

The following example shows a simple OSPF configuration that enables OSPF routing process 1, attaches Ethernet interface 0/0 to Area 0.0.0.0, and redistributes RIP into OSPF and OSPF into RIP:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# ip ospf cost 1
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.17.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0.0.0.0
Router(config-router)# redistribute rip metric 1 subnets
Router(config-router)# exit
Router(config)# router rip
Router(config-router)# network 172.17.0.0
Router(config-router)# redistribute ospf 1
Router(config-router)# default-metric 1
Router(config-router)# !
```

The following example shows the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 1 is initialized, and four OSPF areas are defined: 10.9.50.0, 2, 3, and 0. Areas 10.9.50.0, 2, and 3 mask specific address ranges, whereas Area 0 enables OSPF for all other networks.

```
Router(config)# router ospf 1
Router(config-router)# network 172.18.20.0 0.0.0.255 area 10.9.50.0
Router(config-router)# network 172.18.0.0 0.0.255.255 area 2
Router(config-router)# network 172.19.10.0 0.0.0.255 area 3
Router(config-router)# network 0.0.0.0 255.255.255.255 area 0
Router(config-router)# exit
Router(config)# ! Ethernet interface 0/0 is in area 10.9.50.0:
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.18.20.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 1/0 is in area 2:
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 2/0 is in area 2:
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.18.2.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 3/0 is in area 3:
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 172.19.10.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 4/0 is in area 0:
Router(config)# interface Ethernet 4/0
Router(config-if)# ip address 172.19.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 5/0 is in area 0:
```

```
Router(config)# interface Ethernet 5/0
Router(config-if)# ip address 10.1.0.1 255.255.0.0
Router(config-if)# !
```

Each **network** router configuration command is evaluated sequentially, so the specific order of these commands in the configuration is important. The Cisco IOS software sequentially evaluates the *address wildcard-mask* pair for each interface. See the *IP Routing: Protocol-Independent Command Reference* for more information.

Consider the first **network** command. Area ID 10.9.50.0 is configured for the interface on which subnet 172.18.20.0 is located. Assume that a match is determined for Ethernet interface 0/0. Ethernet interface 0/0 is attached to Area 10.9.50.0 only.

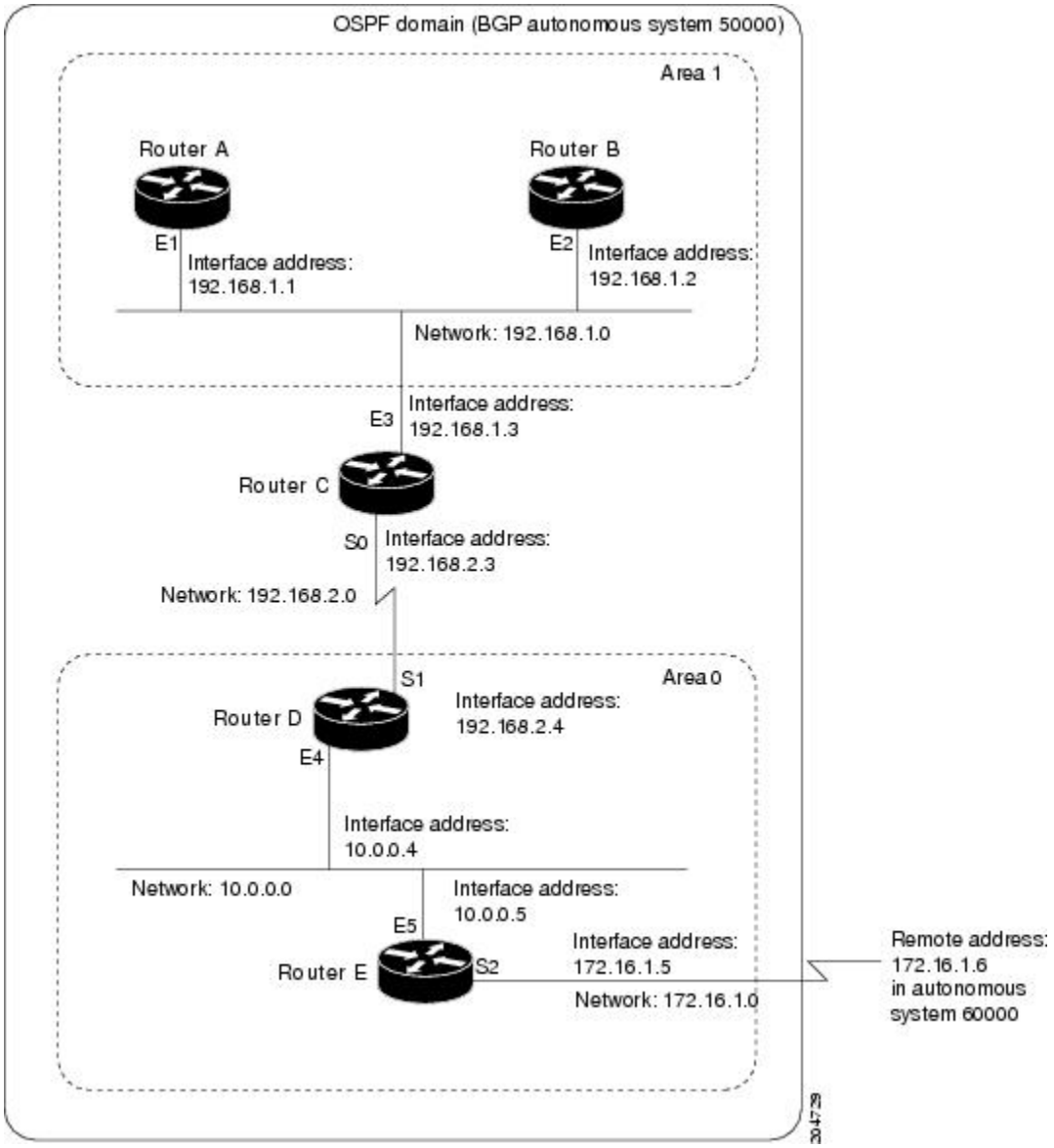
The second **network** command is evaluated next. For Area 2, all interfaces (except Ethernet interface 0/0) are evaluated. Assume that a match is determined for Ethernet interface 1/0. OSPF is then enabled for that interface, and Ethernet 1/0 is attached to Area 2.

This process of attaching interfaces to OSPF areas continues for all **network** commands. Note that the last **network** command in this example is a special case. With this command, all available interfaces (not explicitly attached to another area) are attached to Area 0.

## Example: Internal Router ABR and ASBR Configurations

The figure below provides a general network map that illustrates a sample configuration for several routers within a single OSPF autonomous system.

*Figure 2: Example OSPF Autonomous System Network Map*




In this configuration, the following five routers are configured in OSPF autonomous system 1:

- Router A and Router B are both internal routers within area 1.
- Router C is an OSPF ABR. Note that for Router C, area 1 is assigned to E3 and Area 0 is assigned to S0.
- Router D is an internal router in Area 0 (backbone area). In this case, both **network** router configuration commands specify the same area (Area 0 or the backbone area).
- Router E is an OSPF ASBR. Note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.

**Note** You don't have to include definitions of all areas in an OSPF autonomous system in the configuration of all routers in the autonomous system. You must define only the directly connected areas. In the example that follows, routes in Area 0 are learned by routers in area 1 (Router A and Router B) when the ABR (Router C) injects summary LSAs into Area 1.

Autonomous system 60000 is connected to the outside world via the BGP link to the external peer at IP address 172.16.1.6.

Here is an example configuration for the general network map shown in the figure above.

### Router A Configuration—Internal Router

```
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

### Router B Configuration—Internal Router

```
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 192.168.1.2 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

### Router C Configuration—ABR

```
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 192.168.1.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface Serial 0
Router(config-if)# ip address 192.168.2.3 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# exit
```

### Router D Configuration—Internal Router

```
Router(config)# interface Ethernet 4/0
Router(config-if)# ip address 10.0.0.4 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 1
Router(config-if)# ip address 192.168.2.4 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# exit
```

### Router E Configuration—ASBR

```
Router(config)# interface Ethernet 5/0
Router(config-if)# ip address 10.0.0.5 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 2
Router(config-if)# ip address 172.16.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# redistribute bgp 50000 metric 1 metric-type 1
Router(config-router)# exit
Router(config)# router bgp 50000
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.0.0.0
Router(config-router)# neighbor 172.16.1.6 remote-as 60000
```

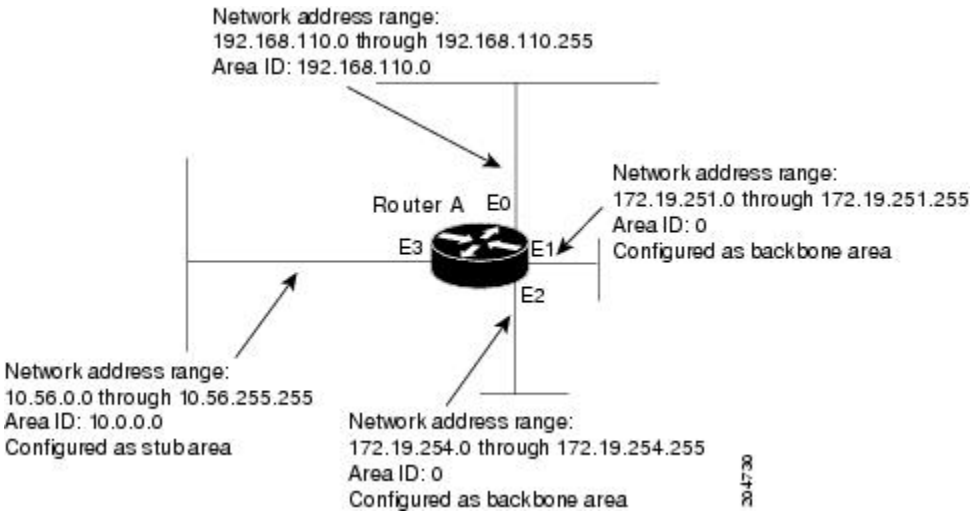
## Example: Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up an ABR. These tasks can be split into the following two general categories:

- Basic OSPF configuration
- Route redistribution

The figure below illustrates the network address ranges and area assignments for interfaces.

***Figure 3: Interface and Area Specifications for the OSPF Configuration***




The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet interface 0 through Ethernet interface 3.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link-state metrics and other OSPF interface configuration options.

- Create a *stub area* with area ID 10.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but they can be merged into a single **area** command.)
- Specify the backbone area (Area 0).

Configuration tasks associated with route redistribution are as follows:

- Redistribute EIGRP and RIP into OSPF with various options set (including **metric-type**, **metric**, **tag**, and **subnet**).
- Redistribute EIGRP and OSPF into RIP.

The following is a sample OSPF configuration:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 192.168.110.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.19.251.201 255.255.255.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf retransmit-interval 10
Router(config-if)# ip ospf transmit-delay 2
Router(config-if)# ip ospf priority 4
Router(config-if)# exit
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.19.254.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 10.56.0.201 255.255.0.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf dead-interval 80
Router(config-if)# exit
```

In the following configuration, OSPF is on network 172.19.0.0:

```
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 10.0.0.0
Router(config-router)# network 192.168.110.0 0.0.0.255 area 192.68.110.0
Router(config-router)# network 172.19.0.0 0.0.255.255 area 0
Router(config-router)# area 0 authentication
Router(config-router)# area 10.0.0.0 stub
Router(config-router)# area 10.0.0.0 authentication
Router(config-router)# area 10.0.0.0 default-cost 20
Router(config-router)# area 192.168.110.0 authentication
Router(config-router)# area 10.0.0.0 range 10.0.0.0 255.0.0.0
Router(config-router)# area 192.168.110.0 range 192.168.110.0 255.255.255.0
Router(config-router)# area 0 range 172.19.251.0 255.255.255.0
Router(config-router)# area 0 range 172.19.254.0 255.255.255.0
Router(config-router)# redistribute eigrp 200 metric-type 2 metric 1 tag 200 subnets
Router(config-router)# redistribute rip metric-type 2 metric 1 tag 200
Router(config-router)# exit
```

In the following configuration, EIGRP autonomous system 1 is on 172.19.0.0:

```
Router(config)# router eigrp 1
Router(config-router)# network 172.19.0.0
Router(config-router)# exit
Router(config)# ! RIP for 192.168.110.0:
Router(config)# router rip
Router(config-router)# network 192.168.110.0
Router(config-router)# redistribute eigrp 1 metric 1
Router(config-router)# redistribute ospf 201 metric 1
Router(config-router)# exit
```

# Example: Default Metric Values Redistribution

The following example shows how a router in autonomous system 1 is configured to run both RIP and EIGRP. The example advertises EIGRP-derived routes using RIP and assigns the EIGRP-derived routes a RIP metric of 10.

```
Router(config)# router rip
Router(config-router)# default-metric 10
Router(config-router)# redistribute eigrp 1
Router(config-router)# exit
```

# Example: Route Map

The examples in this section illustrate the use of redistribution with and without route maps. Examples from both the IP and Connectionless Network Service (CLNS) routing protocols are given below. The following example shows how to redistribute all OSPF routes into EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 101
Router(config-router)# exit
```

The following example shows how to redistribute RIP routes, with a hop count equal to 1, into OSPF. These routes will be redistributed into OSPF as external LSAs with a metric of 5, metric-type of type 1, and a tag equal to 1.

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

The following example shows how to redistribute OSPF learned routes with tag 7 as a RIP metric of 15:

```
Router(config)# router rip
Router(config-router)# redistribute ospf 1 route-map 5
Router(config-router)# exit
Router(config)# route-map 5 permit
Router(config-route-map)# match tag 7
Router(config-route-map)# set metric 15
```

The following example shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 into BGP with an INTER_AS metric of 5:

```
Router(config)# router bgp 50000
Router(config-router)# redistribute ospf 1 route-map 10
Router(config-router)# exit
Router(config)# route-map 10 permit
Router(config-route-map)# match route-type internal
Router(config-route-map)# match interface serial 0
Router(config-route-map)# set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000; these routes will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
Router(config)# router isis
Router(config-router)# redistribute ospf 1 route-map 2
Router(config-router)# redistribute iso-igrp nsfnet route-map 3

Router(config-router)# exit
Router(config)# route-map 2 permit
Router(config-route-map)# match route-type external
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# route-map 3 permit
Router(config-route-map)# match address 2000
Router(config-route-map)# set metric 30
Router(config-route-map)# exit
```

With the following configuration, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
Router(config)# router rip
Router(config-router)# redistribute ospf 101 route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 1 2
Router(config-route-map)# set metric 1
Router(config-route-map)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 3
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
Router(config)# route-map 1 deny
Router(config-route-map)# match tag 4
Router(config-route-map)# exit
Router(config)# route map 1 permit
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
```

Given the following configuration, a RIP learned route for network 172.18.0.0 and an ISO-IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS Level 2 LSP with a metric of 5:

```
Router(config)# router isis
```

```
Router(config-router)# redistribute rip route-map 1
Router(config-router)# redistribute iso-igrp remote route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# match clns address 2
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Router(config)# clns filter-set 2 permit 49.0001.0002...
```

The following configuration example illustrates how a route map is referenced by the **default-information** router configuration command. This type of reference is called conditional default origination. OSPF will originate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.20.0.0 is in the routing table.

```
Router(config)# route-map ospf-default permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type-2
Router(config-route-map)# exit
Router(config)# access-list 1 172.20.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

# Example: Passive Interface

In OSPF, hello packets are not sent on an interface that is specified as passive. Hence, the router will not be able to discover any neighbors, and none of the OSPF neighbors will be able to see the router on that network. In effect, this interface will appear as a stub network to the OSPF domain. This configuration is useful if you want to import routes associated with a connected network into the OSPF domain without any OSPF activity on that interface.

The **passive-interface** router configuration command is typically used when the wildcard specification on the **network** router configuration command configures more interfaces than is desirable. The following configuration causes OSPF to run on all subnets of 172.18.0.0:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.18.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.2.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.18.3.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0

Router(config-router)# exit
```

If you do not want OSPF to run on 172.18.3.0, enter the following commands:

```
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0
Router(config-router)# passive-interface Ethernet 2
Router(config-router)# exit
```

## Example: Configuring Default Passive Interfaces

The following example shows how to configure network interfaces, set all interfaces that are running OSPF as passive, and then enable serial interface 0/0:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.19.64.38 255.255.255.0 secondary
Router(config-if)# ip address 172.19.232.70 255.255.255.240
Router(config-if)# no ip directed-broadcast
Router(config-if)# exit
Router(config)# interface Serial 0/0
Router(config-if)# ip address 172.24.101.14 255.255.255.252
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# exit
Router(config)# interface TokenRing 0
Router(config-if)# ip address 172.20.10.4 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# ring-speed 16
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface Serial 0/0
Router(config-router)# network 172.16.10.0 0.0.0.255 area 0
Router(config-router)# network 172.19.232.0 0.0.0.255 area 4
Router(config-router)# network 172.24.101.0 0.0.0.255 area 4
Router(config-router)# exit
```

# Example: Policy-Based Routing

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1/0/0 from the source 10.1.1.1 are sent to the device at 172.16.6.6 if the device has no explicit route for the destination of the packet. Packets that arrive from the source 172.17.2.2 are sent to the device at 192.168.7.7 if the device has no explicit route for the destination of the packet. All other packets for which the device has no explicit route to the destination are discarded.

```
Device(config)# access-list 1 permit ip 10.1.1.1
Device(config)# access-list 2 permit ip 172.17.2.2
Device(config)# interface async 1/0/0
Device(config-if)# ip policy route-map equal-access
Device(config-if)# exit
Device(config)# route-map equal-access permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip default next-hop 172.16.6.6
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 20
Device(config-route-map)# match ip address 2
Device(config-route-map)# set ip default next-hop 192.168.7.7
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 30
Device(config-route-map)# set default interface null 0
Device(config-route-map)# exit
```

# Example: Policy Routing with Cisco Express Forwarding

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

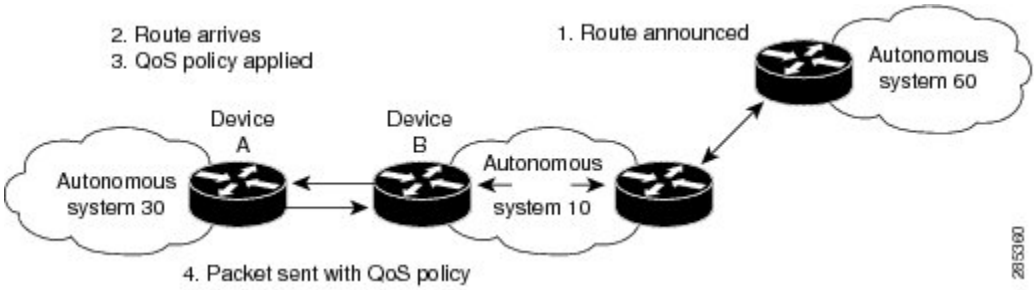
```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip precedence priority
Device(config-route-map)# set ip next-hop 10.0.0.8
Device(config-route-map)# set ip next-hop verify-availability
Device(config-route-map)# exit
Device(config)# route-map test permit 20
Device(config-route-map)# match ip address 101
Device(config-route-map)# set interface Ethernet 0/0/3
Device(config-route-map)# set ip tos max-throughput
Device(config-route-map)# exit
```

# Example: Configuring QoS Policy Propagation via BGP

The following example shows how to create route maps to match access lists, Border Gateway Protocol (BGP) community lists, and BGP autonomous system paths, and apply IP precedence to routes learned from neighbors.

In the figure below, Device A learns routes from autonomous system 10 and autonomous system 60. The quality of service (QoS) policy is applied to all packets that match defined route maps. Any packets from Device A to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps in the figure indicate.

***Figure 4: Device Learning Routes and Applying QoS Policy***




### Device A Configuration

```
interface serial 5/0/0/1:0
ip address 10.28.38.2 255.255.255.0
bgp-policy destination ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF
router bgp 30
```

```
 table-map precedence-map
 neighbor 10.20.20.1 remote-as 10
 neighbor 10.20.20.1 send-community
!
ip bgp-community new-format
!
! Match community 1 and set the IP precedence to priority
route-map precedence-map permit 10
 match community 1
 set ip precedence priority
!
! Match community 2 and set the IP precedence to immediate
route-map precedence-map permit 20
 match community 2
 set ip precedence immediate
!
! Match community 3 and set the IP precedence to flash
route-map precedence-map permit 30
 match community 3
 set ip precedence flash
!
! Match community 4 and set the IP precedence to flash-override
route-map precedence-map permit 40
 match community 4
 set ip precedence flash-override
!
! Match community 5 and set the IP precedence to critical
route-map precedence-map permit 50
 match community 5
 set ip precedence critical
!
! Match community 6 and set the IP precedence to internet
route-map precedence-map permit 60
 match community 6
 set ip precedence internet
!
! Match community 7 and set the IP precedence to network
route-map precedence-map permit 70
 match community 7
 set ip precedence network
!
! Match ip address access list 69 or match autonomous system path 1
! and set the IP precedence to critical
route-map precedence-map permit 75
 match ip address 69
 match as-path 1
 set ip precedence critical
!
! For everything else, set the IP precedence to routine
route-map precedence-map permit 80
 set ip precedence routine
!
! Define community lists
ip community-list 1 permit 60:1
ip community-list 2 permit 60:2
ip community-list 3 permit 60:3
ip community-list 4 permit 60:4
ip community-list 5 permit 60:5
ip community-list 6 permit 60:6
ip community-list 7 permit 60:7
!
! Define the AS path
ip as-path access-list 1 permit ^10_60
!
```

```
! Define the access list
access-list 69 permit 10.69.0.0
```

### Device B Configuration

```
router bgp 10
 neighbor 10.30.30.1 remote-as 30
 neighbor 10.30.30.1 send-community
 neighbor 10.30.30.1 route-map send_community out
!
ip bgp-community new-format
!
! Match prefix 10 and set community to 60:1
route-map send_community permit 10
 match ip address 10
 set community 60:1
!
! Match prefix 20 and set community to 60:2
route-map send_community permit 20
 match ip address 20
 set community 60:2
!
! Match prefix 30 and set community to 60:3
route-map send_community permit 30
 match ip address 30
 set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
 match ip address 40
 set community 60:4
!
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
 match ip address 50
 set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
 match ip address 60
 set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
 match ip address 70
 set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
 set community 60:8
!
! Define access lists
access-list 10 permit 10.61.0.0
access-list 20 permit 10.62.0.0
access-list 30 permit 10.63.0.0
access-list 40 permit 10.64.0.0
access-list 50 permit 10.65.0.0
access-list 60 permit 10.66.0.0
access-list 70 permit 10.67.0.0
```

# Example: Managing Authentication Keys

The following example shows how to configure a key chain named kc1. In this example, the software will always accept and send ks1 as a valid key. The key ks2 will be accepted from 1:30 p.m. to 3:30 p.m. and be sent from 2:00 p.m. to 3:00 p.m. The overlap allows for migration of keys or discrepancy in the set time of the router.

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip rip authentication key-chain kc1
Router(config-if)# ip rip authentication mode md5
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.19.0.0
Router(config-router)# version 2
Router(config-router)# exit
Router(config)# key chain kc1
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string ks1
Router(config-keychain-key)# key 2
Router(config-keychain-key)# key-string ks2
Router(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# key 3
Router(config-keychain-key)# key-string ks3
Router(config-keychain-key)# accept-lifetime 14:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 15:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# exit
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| IPv6 addressing and connectivity | *IPv6 Configuration Guide* |
| IPv6 commands | *Cisco IOS IPv6 Command Reference* |
| Cisco IOS IPv6 features | *Cisco IOS IPv6 Feature Mapping* |

### Standards and RFCs

| Standard/RFC | Title |
|---|---|
| RFCs for IPv6 | *IPv6 RFCs* |

### MIBs

| MIB | MIBs Link |
|---|---|
| | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### 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 Configuring IP Routing Protocol-Independent Features

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 Configuring IP Routing Protocol-Independent Features***