Configure Route Maps Applied with Redistribution Command

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

- Route-Map Basics
- Redistribute OSPF Routes into EIGRP with a Route-Map

Commands Used in Route-Map Applied to Redistribution

- Unsupported Match and Set Commands in Route-Maps
- Source and Target Protocol Behavior in Redistribution Route-Maps

Command Support Tables

Conclusion

Related Information

Introduction

This document describes how to configure route-maps that are applied with the **redistribute** command of dynamic routing protocols.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic IP Routing
- Cisco Command Line Interface (CLI)

Components Used

Although route-map support was originally introduced in Cisco IOS Software, the route-map command remains valid and widely used in Cisco IOS XE. The general route-map concepts, such as ordered sequence evaluation, permit and deny clauses, match statements, and set actions also apply to Cisco IOS XE. Command syntax and feature support can vary by platform, software release, routing protocol, and address family. Some match and set commands supported in one redistribution context can be unsupported or behave differently in another. Always validate the exact command syntax against the target release and platform documentation before deployment.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

This section provides an overview of route-maps used in Cisco IOS software.

Route-Map Basics

Route-maps and access control lists (ACLs) share some common behavior. Both are policy mechanisms that evaluate entries in a defined order and return a permit or deny result.

Common characteristics include:

- **Ordered evaluation:**
  ACLs and route-maps are processed sequentially, from the lowest sequence number to the highest. Evaluation stops when the first matching entry is found, and the action associated with that entry is applied.
- **Permit and deny logic:**
  Both ACLs and route-maps use permit and deny statements. However, the meaning of permit and deny depends on how the ACL or route-map is applied. For example, the effect of a route-map used for redistribution is different from the effect of a route-map used for Policy-Based Routing (PBR) or BGP policy.
- **Context-dependent behavior:**
  ACLs and route-maps are generic policy tools. Their behavior depends on the feature that references them. The same route-map can have different effects when used for redistribution, BGP route policy, PBR, or another feature.

Route-maps and ACLs also have important differences:

- **ACLs provide basic permit or deny matching:**
  An ACL primarily returns a yes or no result. In the context of route redistribution, an ACL or prefix list can be used as match criteria to identify which routes are eligible for redistribution.

- **Route-maps can match and modify routing information:**
  A route-map can reference ACLs, prefix lists, or other match criteria. In addition to permitting or denying routes, route-maps can modify supported route attributes with set commands. For example, a route-map used during redistribution can set a route metric, tag, metric type, or other supported attributes, depending on the source and target routing protocols.

- **Route-maps support more advanced match criteria:**
  Route-maps can match routing attributes that ACLs cannot evaluate directly. For example, a route-map can match route type, route tag, metric, next hop, route source, or other protocol-specific attributes when those match conditions are supported in the redistribution context.

- **End-of-policy behavior depends on the application:**
  ACLs include an implicit deny at the end. Route-map behavior also depends on how the route-map is used. For redistribution, if a route does not match any route-map clause, the route is not redistributed. This functionally is similar to an implicit deny at the end of the route-map.

- **Empty route-map clauses have special significance:**
  A route-map clause without a match statement matches all remaining routes that reach that clause. In redistribution, an empty permit clause permits redistribution of all remaining routes, while an empty deny clause blocks redistribution of all remaining routes.

The dynamic routing protocol redistribute command commonly supports the use of a route-map to control which routes are redistributed and how route attributes are modified during redistribution. When filtering is required based on prefix or mask, configure an ACL or prefix list and reference it from a route-map with a match statement.

Use a route-map in the redistribution process when you must modify route information, such as metric, tag, metric type, or other supported attributes, or when you must match capabilities beyond a simple prefix or mask selection. For simple prefix-based filtering, an ACL or prefix list can still be used, however, it is typically referenced within the route-map that is applied to the redistribute command.

In redistribution, the route-map is applied to routing information selected from the source routing process before that information is injected into the target routing process. It does not directly apply to inbound or outbound packet traffic. Its effect is limited to controlling and modifying redistributed routes.

Redistribute OSPF Routes into EIGRP with a Route-Map

This is a typical Open Shortest Path First (OSPF) to Enhanced Interior Gateway Routing Protocol (EIGRP) route-map, applied with a redistribute command:

<#root>

!

```
router eigrp 1
 default-metric 20000 2000 255 1 1500


redistribute ospf 1 route-map ospf-to-eigrp


!
!
ip prefix-list pfx seq 5 permit 10.0.0.0/8 le 32
!
route-map ospf-to-eigrp deny 10
 match tag 6
 match route-type external type-2
!
route-map ospf-to-eigrp permit 20
 match ip address prefix-list pfx
 set metric 40000 1000 255 1 1500
!
route-map ospf-to-eigrp permit 30
 set tag 8
!
```

Important Observations:

Route-map clauses are numbered. In this example, the clauses have sequence numbers 10, 20, and 30. Sequence numbers allow you to:

- Delete one specific clause without affecting other parts of the route-map.
- Insert a new clause between two existing clauses.

Cisco recommends numbering clauses in intervals of 10 to reserve sequence numbers for future changes, if needed.

Route-maps can have permit and deny clauses. In **route-map ospf-to-eigrp**, there is one deny clause with sequence number 10, and two permit clauses with sequence numbers 20 and 30. The deny clause rejects matching routes from redistribution. Therefore, these rules apply:

- If you use an ACL in a route-map permit clause, routes that are permitted by the ACL are redistributed.
- If you use an ACL in a route-map deny clause, routes that are permitted by the ACL are not redistributed.
- If you use an ACL in a route-map permit or deny clause, and the ACL denies a route, the route-map clause is not considered a match, and the next route-map clause is evaluated.

Each route-map clause can contain two types of commands:

- **match** — Specifies the criteria used to select the routes where the clause applies.
- **set** — Modifies route attributes for the matched routes before they are redistributed into the target routing protocol.

For each redistributed route, the router first evaluates the match commands in a route-map clause. If the match criteria succeeds, the route is redistributed or rejected as dictated by the permit or deny clause. If the route is permitted, its attributes can be modified by the set commands. If the match criteria fails, the clause is not applicable to the route and Cisco IOS software evaluates the route against the next clause in the route-map. The route-map scan continues until a clause is found whose match commands match the route, or until the end of the route-map is reached.

A match or set command can be omitted or repeated in each clause, with these considerations:

- If several match commands are present in a clause, all must succeed for a given route to match the clause. In other words, a logical AND operation is applied to multiple match commands.
- If a single match command refers to several objects, any one of the objects can match. In other words, a logical OR operation is applied within the same match command. For example, with match IP address 101 121, a route matches if it is permitted by access list 101 or access list 121.
- If a match command is not present, all routes match the clause. In the previous example, all routes that reach clause 30 match; therefore, the end of the route-map is never reached.
- If a set command is not present in a route-map permit clause, the route is redistributed without modification of its current attributes.

Do not configure a set command in a deny route-map clause because the deny clause prevents route redistribution. There is no redistributed route information to modify.

A route-map clause without a match or set command still performs an action. An empty permit clause allows redistribution of the remaining routes without modification. An empty deny clause prevents redistribution of the remaining routes. This is also the default behavior if a route-map is completely scanned and no explicit match is found.

Based on the information in this section, the previous OSPF-to-EIGRP route-map example does this:

- Prohibits redistribution of all Type-2 external OSPF routes with tag 6.
- Redistributes into EIGRP all routes that match the pfx prefix list and sets the EIGRP metric values to 40000, 1000, 255, 1, and 1500.
- Redistributes all other routes and sets their tag to 8.

In this example, the topology consists of three routers connected in a linear arrangement: R3 ↔ R1 ↔ R2. Router R1 acts as the redistribution router between R3 and R2, with no direct connection between R3 and R2. Each router is connected using GigabitEthernet interfaces, providing high-speed point-to-point connectivity between the adjacent devices.

<#root>

R3#

show ip route 10.20.20.20

Routing entry for 10.20.20.20/32
  Known via "eigrp 1",

distance 170

, metric 320256, precedence routine (0), type

external

  Redistributing via eigrp 1
  Last update from 192.168.1.1 on GigabitEthernet1, 00:01:02 ago
  Routing Descriptor Blocks:
  * 192.168.1.1, from 192.168.1.1, 00:01:02 ago, via GigabitEthernet1
      Route metric is 320256, traffic share count is 1

Total delay is 10010 microseconds, minimum bandwidth is 40000 Kbit Reliability 255/255, minimum MTU 150

      Loading 1/255, Hops 1

R3#

show ip route 172.16.1.0

Routing entry for 172.16.1.0/24
  Known via "eigrp 1",

distance 170

, metric 640256
  Tag 8, precedence routine (0), type

external

  Redistributing via eigrp 1
  Last update from 192.168.1.1 on GigabitEthernet1, 00:01:56 ago
  Routing Descriptor Blocks:
  * 192.168.1.1, from 192.168.1.1, 00:01:56 ago, via GigabitEthernet1
      Route metric is 640256, traffic share count is 1

Total delay is 20010 microseconds, minimum bandwidth is 20000 Kbit Reliability 255/255, minimum MTU 150

      Loading 1/255, Hops 1

Route tag 8

R3#

Running the **show ip route <prefix>** command on R3 to confirm how each redistributed route was processed by the route-map. The route 10.20.20.20/32 matches route-map sequence 20 because it is permitted by the pfx prefix-list. As a result, the route is redistributed into EIGRP with the metric explicitly configured under sequence 20 with the **set metric 40000 1000 255 1 1500** command.

These characteristics are reflected in the route output:

- The route is learned as an EIGRP external route.
- The minimum bandwidth is 40000 Kbit.
- The total delay is 10010 microseconds.
- No route tag is shown because sequence 20 only sets the metric and does not set a tag.

Additionally, route 172.16.1.0/24 does not match sequence 10 or sequence 20, so it reaches route-map sequence 30. Since sequence 30 has no match statement, it matches all remaining routes and applies the set tag 8 that is reflected in the route output:

- The route is learned as an EIGRP external route.
- The route includes Tag 8.
- The output also shows Route tag 8 under the route descriptor.
- The metric uses the redistribution default metric configured under the EIGRP process, rather than the metric from sequence 20.

These outputs confirm the route-map processes routes in order: sequence 20 applies the explicit EIGRP metric to prefixes matched by the prefix list, while sequence 30 permits the remaining redistributed routes and marks them with tag 8.

Although the physical links between R3, R1, and R2 are GigabitEthernet interfaces with DLY 10 usec, the redistributed EIGRP routes also carry the seed delay configured during redistribution. In the set metric and default-metric commands, the EIGRP delay value is entered in tens of microseconds. Therefore, a configured delay of 1000 becomes 10000 microseconds, and a configured delay of 2000 becomes 20000 microseconds. When R3 learns the routes from R1, it adds the local GigabitEthernet interface delay of 10 microseconds, which results in total delays of 10010 microseconds and 20010 microseconds, respectively.

Commands Used in Route-Map Applied to Redistribution

This section includes these topics:

- Configure Unsupported Match and Set Commands in Route-Maps
- Two-Protocol Nature of Route-Map Redistribution
- Command Support Tables

Unsupported Match and Set Commands in Route-Maps

Route-maps are generic mechanisms you can use in many configuration contexts, including the previously described redistribute command. You can configure and run the **match length** command in a route-map used for Policy-Based Routing (PBR) to specify a particular action is taken when packets of a specific length are forwarded. However, the **match length** command is not applicable in route-maps used for route redistribution.

You can configure and run **match** and **set** commands in a route-map that are not supported, or have no effect in the context where the route-map is applied. An example is running the **match length** command in a route-map applied to redistribution. In redistribution, the route-map is applied to routes installed in the routing table by the source protocol specified in the redistribute command. Therefore, when the router processes the route-map, it interprets only the commands that are meaningful in the redistribution context. In this example, the **match length** command has no effect on redistribution. The command remains in the route-map configuration and appears in the running configuration, however, the route redistribution is not affected by whether the command is present.

Although the router can allow different types of **match** and **set** commands to be configured under a route-map, those commands must be appropriate for the feature or protocol context where the route-map is used. Otherwise, the configuration can become confusing or produce unintended results.

Do not use commands that have no effect in a specific route-map context, even if they appear harmless, for these reasons:

- Commands with no effect can obscure the intended policy and cause confusion.
- Commands that are unsupported in the current context can become supported in a future Cisco IOS or Cisco IOS XE release, which can cause unexpected changes in route-map behavior after a software upgrade.
- Not all unsupported commands are harmless. For example, the set metric +/- command specifies a relative metric change and is used in some Border Gateway Protocol (BGP) route advertisement context. It can take the current metric of a route and increase or decrease it by the specified value before the route is propagated.



Caution: The **set metric +<value>** and **set metric -<value>** syntax is not supported by all routing protocols. On platforms or protocols that do not support the + or - form, the sign can be ignored and the command interpreted as **set metric <value>**. This can result in an unintended metric being assigned during redistribution, potentially affecting route selection and network traffic. Always verify the platform and protocol support before using the incremental or decremental metric syntax.



Note: Refer to EIGRP Route Map Support for additional information.

For example, consider this route-map:

<#root>

! route-map ospf-to-ospf permit 10

set metric +2

 !

This configuration appears to redistribute all routes from one OSPF process to another, while it increases the metric of all routes by two. It sets the metric of all routes to be the same, equal to 2. This is unexpected in the router configuration.

<#root>

R2#

show ip route

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 2 subnets
C        10.20.20.20 is directly connected, Loopback0
```

O E2 10.30.30.30 [110/2] via 172.16.1.1, 00:00:26, GigabitEthernet1

```
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/24 is directly connected, GigabitEthernet1
L        172.16.1.2/32 is directly connected, GigabitEthernet1
```

O E2 192.168.1.0/24 [110/2] via 172.16.1.1, 00:00:26, GigabitEthernet1

R2#

show ip route 10.30.30.30

```
Routing entry for 10.30.30.30/32
  Known via "ospf 1", distance 110, metric 2, type extern 2, forward metric 1
  Last update from 172.16.1.1 on GigabitEthernet1, 00:00:38 ago
  Routing Descriptor Blocks:
  * 172.16.1.1, from 192.168.1.1, 00:00:38 ago, via GigabitEthernet1
```

Route metric is 2

```
, traffic share count is 1
```

```
R2#
```

show ip route 192.168.1.0

```
Routing entry for 192.168.1.0/24
  Known via "ospf 1", distance 110, metric 2, type extern 2, forward metric 1
  Last update from 172.16.1.1 on GigabitEthernet1, 00:01:34 ago
  Routing Descriptor Blocks:
  * 172.16.1.1, from 192.168.1.1, 00:01:34 ago, via GigabitEthernet1
```

Route metric is 2

```
, traffic share count is 1
```

This route-map provides a counter-intuitive effect:

<#root>

!
route-map ospf-to-ospf permit 10

set metric -367

```
!
```

Rather than decrease the metric of redistributed routes, this configuration actually sets the metric to 367 (a positive value, because a negative metric is not possible when **set metric** is interpreted without the sign).

<#root>

R2#

show ip route

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 2 subnets
C        10.20.20.20 is directly connected, Loopback0

O E2 10.30.30.30 [110/367] via 172.16.1.1, 00:00:06, GigabitEthernet1


      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/24 is directly connected, GigabitEthernet1
L        172.16.1.2/32 is directly connected, GigabitEthernet1

O E2 192.168.1.0/24 [110/367] via 172.16.1.1, 00:00:06, GigabitEthernet1



R2#

show ip route 10.30.30.30


Routing entry for 10.30.30.30/32
  Known via "ospf 1", distance 110, metric 367, type extern 2, forward metric 1
  Last update from 172.16.1.1 on GigabitEthernet1, 00:00:17 ago
  Routing Descriptor Blocks:
  * 172.16.1.1, from 192.168.1.1, 00:00:17 ago, via GigabitEthernet1



Route metric is 367


, traffic share count is 1

R2#

show ip route 192.168.1.0


Routing entry for 192.168.1.0/24
  Known via "ospf 1", distance 110, metric 367, type extern 2, forward metric 1
  Last update from 172.16.1.1 on GigabitEthernet1, 00:00:31 ago
  Routing Descriptor Blocks:
  * 172.16.1.1, from 192.168.1.1, 00:00:31 ago, via GigabitEthernet1



Route metric is 367


, traffic share count is 1
```

Source and Target Protocol Behavior in Redistribution Route-Maps

Route-maps applied to redistribution work with two routing protocols:

- The **source protocol**, which provides the original routing information.
- The **target protocol**, which the routing information is redistributed.

Each routing protocol supports its own set of route attributes. For this reason, match and set commands in a redistribution route-map are evaluated in relation to different protocols:

- Route-map match commands verify attributes of routes learned from, or installed by, the source protocol specified in the redistribute command.
- Route-map set commands modify route attributes that are supported by the target protocol where the routes are redistributed.

The Command Support Tables section lists supported commands. These are categorized as **match** and **set** commands to show how redistribution route-maps operate across the source and target routing protocols.

Command Support Tables

This section describes the commands that are supported in route-maps attached in the **redistribute** command. There are seven routing protocols that routes can be redistributed; however, there are only five where redistribution takes place. Connected and static routes are not dynamic routing protocols and can only provide information to be redistributed into other protocols.

Intermediate System-to-Intermediate System (IS-IS) and BGP can propagate information about Connectionless Network Service (CLNS) routes along with IP routes. To be thorough, the tables in this section also mention CLNS-related commands, which can be used in redistribution route-maps for these protocols.

You can use Routing Information Protocol (RIP), OSPF, IS-IS, and BGP to propagate IPv6 routes; redistribution route-maps for these protocols can contain IPv6-specific commands. The **match ip** and **set ip** commands are specific for redistribution of IPv4 prefixes. The **match ipv6** and **set ipv6** commands are specific for redistribution of IPv6 prefixes. You can use the **match clns** and **set clns** commands only if you use a route-map to redistribute CLNS routes to or from the routing protocol.

Table 1 and Table 2 use these conventions:

- Supported commands are marked with a**Yes**.

- Unsupported commands are marked with a dash (—).
- Unsupported commands known to perform an action (probably an unwanted action) are marked with **No**.

Table 1 – Route-map Commands to Match Routes Installed into Routing Tables by Protocols

Command	Redistribution Support						
	Connected	**Static**	**RIP**	**EIGRP**	**OSPF**	**IS-IS**	**BGP**
match clns address	—	Yes	—	—	—	Yes	Yes
match clns next-hop	—	Yes	—	—	—	Yes	—
match interface	Yes	Yes	Yes	Yes	Yes	Yes	—
match ip address	Yes	Yes	Yes	Yes	Yes	Yes	Yes
match ip address prefix-list	Yes	Yes	Yes	Yes	Yes	Yes	Yes
match ip next-hop	—	Yes	Yes	Yes	Yes	Yes	Yes
match ip next-hop prefix-list	—	No	No	No	No	No	No
match ip route-source	—	—	Yes	Yes	Yes	—	Yes
match ip route-source prefix-list	—	—	No	No	No	—	No
match ipv6 address [prefix-list]	Yes	Yes	Yes	—	Yes	Yes	Yes
match ipv6 next-hop [prefix-list]	—	Yes	Yes	—	—	—	Yes
match ipv6 route-source [prefix-list]	—	—	Yes	—	—	—	Yes
match metric	—	—	Yes	Yes	Yes	Yes	Yes
match policy-list	Yes	Yes	Yes	Yes	Yes	Yes	Yes

match route-type external	—	—	—	Yes	Yes	Yes	Yes
match route-type internal	—	—	—	Yes	Yes	—	Yes
match route-type local	—	—	—	—	—	—	Yes
match route-type nssa-external	—	—	—	—	Yes	—	—
match route-type {level-1|level-2}	—	—	—	—	—	Yes	—
match tag	—	Yes	Yes	Yes	Yes	Yes	Yes

Table 2 – Commands that Modify Route Attributes During Redistribution into the Target Protocol

Command	**Redistribution Support**				
	RIP	**EIGRP**	**OSPF**	**IS-IS**	**BGP**
set as-path tag	—	—	—	—	Yes
set community	—	—	—	—	Yes
set ip next-hop	—	—	—	—	Yes
set ip next-hop peer-address	—	—	—	—	No
set ipv6 next-hop	—	—	—	—	Yes
set level {backbone|stub-area}	—	—	No	—	—
set level {level-1|level-2|level-1-2}	—	—	—	Yes	—
set local-preference	—	—	—	—	Yes
set metric	Yes	—	Yes	Yes	Yes

set metric +/-	No	—	No	No	No
set metric eigrp-metric	—	Yes	—	—	—
set metric +/- eigrp-metric	—	No	—	—	—
set metric-type internal	—	—	—	Yes	—
set metric-type external	—	—	—	Yes	—
set metric-type {type-1|type-2}	—	—	Yes	—	—
set nlri	—	—	—	—	Yes
set origin	—	—	—	—	Yes
set tag	Yes	Yes	Yes	—	—
set weight	—	—	—	—	Yes

Conclusion

Route maps are powerful and flexible tools for controlling route redistribution. They provide granular control over how routing information is exchanged between routing protocols during redistribution. When implementing mutual redistribution, use route tags and explicit filtering policies to prevent redistributed routes from being reintroduced into their originating routing protocol. Failure to do so can result in routing loops, traffic blackholes, or suboptimal forwarding paths. Because redistribution between multiple routing protocols can significantly increase routing complexity, networks must be carefully designed, implemented, and validated before deploying advanced redistribution policies.

Related Information

- Redistribution of Routing Protocols
- Protocol Support Page
- Cisco Technical Support & Downloads