Route Mapping

Teldat-Dm 764-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat-Dm 745-I Policy Routing

Teldat-Dm 763-I BGP Protocol

Chapter 1 Introduction

1.1 Route maps

In some complex routing scenarios, applying a dynamic routing protocol is simply not enough. It is necessary to have several routing protocols interoperating, exchanging routes, defining the routes that should be broadcast (and the manner to do it), or implementing static routing with more criteria than the destination IP address. To tackle these scenarios, we need to define routing policies that accurately specify route management in the routing protocols.

Routing policies are defined through route-maps. A route-map is made up of various entries that determine when to act (match clauses) and what actions to take (set clauses). Once the route-maps are defined, various functionalities (such as policy routing, route re-distribution, route filtering and route broadcast control) can be applied.

Currently, route-maps are applied in the functionalities described in the following sections.

1.1.1 Policy routing

The "policy routing" functionality allows you to select which traffic to route and how. The commands applying route-maps to policy routing can be found in the IP protocol configuration menu and in the IP parameter configuration from the interfaces configuration menus where these are applied, and are as follows:

```
IP config>local policy route-map <idmap>
<interface_name> config>ip policy route-map <idmap>
```

For further information on these commands, please see manual Teldat-Dm745-I Policy Routing.

1.1.2 Route Redistribution

You can use route-maps to define which routes are distributed among the protocols and how.

Route-maps are applied through the redistribute command (found in the configuration menu of the routing protocol we want to use to redistribute routes).

```
RIP config>redistribute {bgp | connected | ospf | static} routemap <idmap>
OSPF config>redistribute {bgp | connected | rip | static} routemap <idmap>
```

For further information on these commands, please see the manual on the appropriate routing protocol.

1.1.3 Exchanging Routes in BGP

BGP protocol allows you to specify which routes are exchanged with the neighbors in BGP connections and what attributes are applied. To do this, apply the required route-map in the BGP connections configuration, both in the entries (reception of neighbor routes) and exits (communication of routes to the neighbor). The commands involved here are configured in the BGP group menu and are as follows:

```
Group config>option in-route-map <idmap>
Group config>option out-route-map <idmap>
Group config>peer <peer-address> in-route-map <idmap>
Group config>peer <peer-address> out-route-map <idmap>
```

For further information on these commands please see manual Teldat-Dm763-I BGP Protocol.

1.2 Route-Map Elements

A route-map is made up of a list of ordered entries including selection criteria (match clauses) and actions to carry out (set clauses). Optionally, there may be other modifiers that alter the way the entry is applied. All entries shaping a route-map are grouped under the same name or identifier chosen by the user.

The following sections detail the functions of route-map entries.

1.2.1 Route-Map Entries

Each entry in a route-map contains a combination of match and configuration clauses / commands. Match clauses define the packet or route selection criteria. These must comply with the specified policy (i.e. the conditions to be fulfilled) to be processed according to the pertinent entry. The configuration clauses determine how to process the packets once they coincide with the match criteria.

For each match and configuration command combination in a route-map, the packet or route *must simultaneously comply with all the match clauses* so the configuration clauses can be applied. A route-map can contain multiple groups of configuration and match command combinations (i.e. multiple entries).

Route-map entries can also be configured as allowed (permit) or denied (deny). If the entry is configured as deny, the packets or routes matching the criteria (match clauses) are treated as if they do not coincide with any route-map entry. Entry configuration clauses are applied only where the entry is configured as permitted and meets all the match criteria.

Optionally, you can associate an entry to an external element that sets when you must apply said entry and when you must omit it.

Route-map entries are applied in numerical order (marked by the identifier) until a matching one is found. If no entry matches, this acts in the same way as if the entry was configured as deny.

Note

A deny entry must be inserted at the end of the list of entries.

1.2.2 Match Clauses – Criteria Definition

The following sections detail the match criteria that can be applied to an entry.

1.2.2.1 Access Control Lists to select packets

You can specify as many standard as extended or stateful access control lists. These lists are used to check the packets going to be routed and determine if they are processed according to this entry or not.

There are two different cases: one for IPv4 packets and another for IPv6 packets:

IPv4:

```
Route map id-map>entry <n> match ip address <acl> [...<acl>]
```

IPv6:

You can only specify one IPv6 access list for each route-map.

```
Route map id-map>entry <n> match ipv6 address <ipv6-acl>
```

1.2.2.2 Access Control Lists to select routes

You can specify standard access control lists to select routes. On checking a route against a standard access control list, the route destination network is compared with the access list configuration.

There are two different cases: one for IPv4 routes and another for IPv6 routes.

IPv4:

```
Route map id-map>entry <n> match ip address <acl> [...<acl>]
```

IPv6:

You can only specify one IPv6 access list for each route-map.

```
Route map id-map>entry <n> match ipv6 address <ipv6-acl>
```

1.2.2.3 Prefix lists for selected routes

Prefix lists are more appropriate for selecting routes in a Route Map. Just like with an access list, the route destination network is matched against the configuration of the prefix list.

```
Route map id-map>entry <n> match ip prefix-list <prl>
```

1.2.2.4 Route Next-Hop

There are two different cases: one for IPv4 routes and another for IPv6 routes.

You can specify standard access control lists to select routes with a fixed next hop. If the next hop is an IP address, this is directly checked against the access list. If the next hop is an interface, IP address 0.0.0.0 is checked against the access list or, in the case of IPv6 routes, with an "unspecified" (::) IPv6 address.

IPv4:

```
Route map id-map>entry <n> match ip next-hop <acl> [...<acl>]
```

IPv6:

You can only specify one IPv6 access list for each route-map.

```
Route map id-map>entry <n> match ipv6 next-hop <ipv6-acl>
```

1.2.2.5 Packet Length

You can specify the maximum and minimum length a packet can have so the entry configuration clauses may be applied. The network manager can use the packet length match as the criteria to distinguish between interactive and volume traffic (volume traffic is usually made up of bigger packets).

```
Route map id-map>entry <n> match length <min> <max>
```

1.2.2.6 Route-Source

In OSPF route filtering, you can specify control access lists in order to select the routes that have a given source. This checks if the Router ID announcing the route matches the access control list.

```
Route map id-map>entry <n> match ip route-source <acl>
```

1.2.2.7 Route output interface

You can specify interfaces for route output. If the route's next hop is through one of the configured interfaces, the route-map entry is selected.

```
Route map id-map>entry <n> match interface <ifc>
```

1.2.2.8 Route metric

You can use a given metric value as selection criteria to configure the matching.

```
Route map id-map>entry <n> match metric <value>
```

1.2.2.9 Type of route

In order to select the OSPF routes, you can configure the type of route: internal (intra-area and inter-area) external, type 1 external or type 2 external.

```
Route map id-map>entry <n> match route-type {external | internal | external-type-1 |
external-type-2}
```

1.2.2.10 Route tag

External OSPF routes have an associated Tag value. This value can be used to select the routes through a route-map to filter OSPF routes.

```
Route map id-map>entry <n> match tag <val>
```

1.2.2.11 Route Communities

You can specify a group of BGP communities so that any route that does not belong to one of them fails the test (i.e. the route must belong to one of the configured communities to comply with the selection criteria).

```
Route map id-map>entry <n> match community <community>
```

1.2.2.12 External entities tracing

In addition to the match criteria used by route maps, you can associate the entry to an external entity (i.e. an NSLA advisor) so that the entry is only applied if the advisor is active (TRUE). In cases where the linked advisor is deactivated (FALSE) this entry is ignored, just as if there was not match.

```
Route map id-map>entry <n> track nsla-advisor <advisor-id>
```

1.2.2.13 Origin of a BGP route

Allows you to filter the routes based on their origin. The origin is an attribute of the BGP routes themselves, indicating where they have originated from. The values allowed [RFC4271] are as follows:

	Description
(0) IGP	The route has originated in the initial Autonomous System
(1) EGP	The route has been learnt through the EGP protocol.
(2) INCOMPLETE	The origin of the route is unknown.

1.2.3 Configuration Clauses

When the match clauses in a permitted entry have been satisfied, the configuration clauses are applied. We will now list the configuration clauses that can be applied.

1.2.3.1 Next Hop

There are two different cases: one for IPv4 packets and another for IPv6 packets.

IPv4:

List of IP addresses and/or interfaces through which packets can be routed. You can also specify *local* so the packet is processed as if the device itself was the destination. The IP address can specify the next hop router towards the destination where the packets must be routed. To send packets, the first of the two following conditions that is active is used: the first IP address associated with an interface currently connected or the first specified interface found in an ‘up’ state or the *local* clause.

```
Route map id-map>entry <n> set ip next-hop {<ip-address> | <interface> | local |
peer-address} [...<ip-address> | ...<interface>]
```

IPv6:

```
Route map id-map>entry <n> set ipv6 next-hop {<ipv6-address> | peer-address}
[...<ipv6-address>]
```

If the Route Map is applied to the BGP protocol, with this configuration clause, the next hop for the imported route or the route going to be exported is modified. Whenever the *peer-address* has been specified as the next hop, the address of the BGP peer we received the route from (for imported routes) and the address of the BGP peer that is sending the route (for exported routes) is used.

It must be pointed out that you cannot configure *peer-address* as next hop with any of the other options. This option is only used with the BGP protocol.

1.2.3.2 Default Next Hop

List of default IP addresses and/or interfaces. You can also specify *local* so the packet is processed as if the device itself was the destination. Routing towards the interface or the next hop specified by this configuration clause when no other specified route has been found for the packet destination address in the routing table. To send packets, the first of the two following conditions that is active is used: the first IP address associated with an interface currently connected or the first specified interface found in an ‘up’ state or the *local* clause.

```
Route map id-map>entry <n> set ip default next-hop {<ip-address> | <interface> |
local} [...<ip-address> | ...<interface>]
```

1.2.3.3 Type of IP Service

Type of IP Service (IP TOS). You can specify a value or password to configure the type of service in IP packets.

```
Route map id-map>entry <n> set ip tos <type-of-service>
```

1.2.3.4 IP Precedence

IP Precedence. You can specify a value or password to establish precedence in IP packets.

```
Route map id-map>entry <n> set ip precedence <precedence>
```

1.2.3.5 DSCP-IP

DSCP field from the IP service (IP differentiated services codepoint). A value or a password can be specified to establish the DSCP field in IP packets.

```
Route map id-map>entry <n> set ip dscp <dscp>
```

1.2.3.6 IP don't fragment bit

DF IP (Don't Fragment). You can specify a value or password to establish the DF field in IP packets

```
Route map id-map>entry <n> set ip df [<df>]
```

1.2.3.7 IP service octet

This includes all the byte whose destination is the IP header TOS. You can simply specify a value or add a mask indicating which bits you want this to act over.

```
Route map id-map>entry <n> set ip tos-octet <tos-octet> [mask <mask>]
```

1.2.3.8 Community

Community attributes to establish in BGP routes.

```
Route map id-map>entry <n> set community <community>
```

1.2.3.9 Label

Label value for the packet to exchange. The label is an internal parameter, associated to each IP packet, used to filter and classify IP traffic.

```
Route map id-map>entry <n> set label <value>
```

1.2.3.10 Local Preference

Local Preference attribute to establish in BGP routes.

```
Route map id-map>entry <n> set local-preference <preference>
```

1.2.3.11 Metric

Metric (cost) to establish on redistributing one routing protocol route to another.

This is also used to establish the metric in BGP routes. This metric is broadcast in BGP sessions through the MULTI_EXIT_DISC attribute.

```
Route map id-map>entry <n> set metric <metric>
```

1.2.3.12 Metric Type

Type of OSPF route to establish on redistributing other protocol routes to OSPF.

```
Route map id-map>entry <n> set metric-type {type-1 | type-2}
```

1.2.3.13 Modifying the autonomous system path (AS-PATH attribute) for BGP routes

Sometimes it may be necessary to modify a BGP AS-PATH attribute to influence the BGP decision process (please see manual Teldat-DM763-I “BGP Protocol). This can be done by using the following command in a route-map.

```
Route map id-mapa>entry <n> set as-path prepend <as-number>
```

This command puts the number of the autonomous system (AS) at the front, specifying the autonomous systems path to a BGP route. You can put as many AS numbers as you wish and you can repeat them. However, please bear in mind that they will be added to the path in the order they have been configured. This clause only applies if the route is exported to an external autonomous system.

1.2.3.14 Origin of a route

The “origin” attribute to be established in the BGP routes. The values allowed are the same as in the matching clause case.

1.2.3.15 AFS session mark

Session mark of an Advanced Firewalling System (AFS) session. This is an internal parameter, associated to the session that can be used as a selection criteria in an access-list, and exported through Netflow/IPFIX.

```
Route map id-map>entry <n> set ip session-mark <0..65535>
```

1.2.3.16 AFS AppID

AppId of an Advanced Forwarding System (AFS) session. Custom AppId has the format of a USER-defined (6) classification engine, followed by the custom application ID. The AppId is a parameter of AFS sessions that can be used as a selection criteria in stateful access-lists, and exported through Netflow/IPFIX.

```
Route map id-map>entry <n> set app-id custom id <0..65535>
```

Chapter 2 Configuring Route-Maps

2.1 Introduction

To access the route-map configuration menu, use the **feature route-map** command (found in the main menu of the configuration console).

```
*config
Config>feature route-map

-- Route maps user configuration --
Route map config>
```

Once you are in the route-maps configuration menu, use the configuration commands explained in the following chapter to define the route-maps used in Policy Routing, route redistribution, etc.

2.2 Configuration Commands

The route-map configuration menu has the following available commands:

2.2.1 list

This command displays a list with the names of all defined route-maps.

Syntax:

```
Route map config>list
```

Example:

```
Route map config>list

Configured route maps:
  route-map admin
  route-map office
Route map config>
```

2.2.2 no route-map

This command erases a route-map definition.

Syntax:

```
Route map config>no route-map <id-map>
```

id-map	Name of the route-map to delete.

Example:

```
Route map config>no route-map office
Route map config>
```

2.2.3 route-map

This command enters the configuration mode for the defined route-map.

Syntax:

```
Route map config>route-map <id-map>
```

id-map	Name of route-map to define.

Example:

```
Route map config>route-map office
Route map office>
```

A route-map is made up of one or more numerically ordered entries.

When a route-map is processed, the packet or route is successively checked against each entry until a match is found (complies with the match clauses).

If no entry matching the packet is found, the process applied is the same as if the packet matches a denied entry (configured as deny). The meaning of this depends on the functionality applied to the route-map, therefore:

- Policy routing: the packet is routed in accordance with the routing table.
- Route redistribution: the route is not redistributed.
- OSPF route filtering: the route doesn't install in the active routing table.
- BGP connections: The route is not broadcast. If this applies to the output (out-route-map), the route is not communicated to the BGP connection neighbor. If this is applied to an entry (in-route-map), the received route is discarded and consequently not entered in the routing table.

If an entry matching the packet is found and this is in permit mode (default mode), the configuration clauses (set) are applied to said entry.

The route-map configuration menu has the following available commands:

2.2.3.1 entry <n> default

This command establishes the default configuration for the specified entry.

Syntax:

```
Route map <map-id>>entry <n> default
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

Example:

```
Route map office>entry 1 default
Route map office>
```

2.2.3.2 entry <n> deny

This command establishes the deny mode in an entry. Packets matching this entry are conventionally routed through the routing table and no set clauses are applied to them. Routes that match this entry are neither redistributed nor broadcast.

Syntax:

```
Route map <map-id>>entry <n> deny
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

Example:

```
Route map office>entry 1 deny
Route map office>
```

2.2.3.3 entry <n> permit

This command establishes the permit mode in an entry. Packets matching this entry have set clauses applied to them. This is the default mode for an entry.

Syntax:

```
Route map <map-id>>entry <n> permit
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

Example:

```
Route map office>entry 1 permit
Route map office>
```

2.2.3.4 [no] entry <n> match ip address

This command determines the access lists that must be used to find out if a packet or a route complies with the entry.

In Policy Routing, the packet is checked against the configured access lists.

In route Redistribution, when route filtering is done in OSPF and BGP, this command checks the routes destination network against the configured access lists.

The access lists associated through this command must have been previously created and cannot be associated with any other protocol.

The access lists are checked in the order they were linked to this command.

If a packet or route does not match an access list, or matched it but this is *deny*, the next access list configured in the entry is checked.

If the packet or route matches an access list and it is in *permit* mode, the remaining access lists are not checked and the condition configured in the entry is considered fulfilled.

Therefore, if the packet or route does not match any access list that is *permit*, said packet or route is considered as not matching the route map entry. Thus, the next route-map entry is checked.

Syntax:

```
Route map <map-id>>entry <n> match ip address {<access-list>} [...<access-list>]
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.
access-list	Number of the access control list. You can specify as many standard as extended or stateful lists for Policy Routing. You only need to specify standard lists for route checking. The range goes from 1 to 9999.

Example:

```
Route map office>entry 1 match ip address 101
Route map office>
```

2.2.3.5 [no] entry <n> match ip prefix-list

This command determines the prefix list or list to be used when deciding if a route matches the route map entry.

This clause is only applied in Route Redistribution, route filtering in OSPF and in BGP. Through this you can check the destination networks for the routes with the configured prefix lists.

The associated prefix list must have been previously created, as they can be assigned to other protocols (the opposite of what happens with access lists).

The prefix lists check the order in which they have been associated with this command.

If the route does not match a prefix list, or it matches one but it is *deny*, the next prefix list configured in the entry is checked.

If the route matches a prefix list and the latter is *permit*, no further checking is carried since the condition configured in the entry is considered to be fulfilled.

If the route does not match any prefix list that is *permit*, the route is considered as not matching the route map entry and the next route map entry is checked.

Syntax:

```
Route map <map-id>>entry <n> match ip prefix-list {<prefix-list>}
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.
Prefix-list	Prefix list number. The range goes from 1 to 199.

Example:

```
Route map office>entry 1 match ip prefix-list 1
Route map office>
```

2.2.3.6 [no] entry <n> match ip next-hop

This command determines the access lists to be used to find out if the next hop for a route matches the entry.

This clause is only applied to route Redistribution, BGP and OSPF route filtering.

The access lists associated through this command must have been previously created and cannot be associated with any other protocol.

The access lists are checked in the order in which they were linked to this command.

If the route does not match an access list, or matches one classified as *deny*, the next access list configured in the entry is checked.

If the route matches an access list in *permit* mode, the remaining access lists are not checked since the condition configured in the entry is considered fulfilled.

Therefore, if the route does not match any access list classified as *permit*, said packet or route is considered as not matching the route map entry. The next route-map entry is subsequently checked.

Syntax:

```
Route map <map-id>>entry <n> match ip nexthop {<access-list} [...<access-list>]
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

access-list Number of the access control list. You can only specify standard lists. The range goes from 1 to 99.

Example:

```
Route map office>entry 1 match ip next-hop 1
Route map office>
```

2.2.3.7 [no] entry <n> match ip route-source

This command determines the access lists to be used when finding out if the Router ID for the OSPF route source matches the entry.

This clause is only applied to OSPF route filtering.

The access lists associated to this command must be previously created and cannot be associated to any other protocol.

The access lists are checked in the order in which they were linked to this command.

If the route does not match an access list, or matches one classified as *deny*, the next access list configured in the entry is checked.

If the route matches an access list classified as *permit*, no more access lists are checked as the condition configured in the entry is considered fulfilled.

Therefore, if the route does not match any access list classified as *permit*, said packet or route is considered as not matching the route map entry. The next route-map entry is subsequently checked.

Syntax:

```
Route map <map-id>>entry <n> match ip route-source <accesslist>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

access-list Number of the access control list. Only standard lists can be specified. The range goes from 1 to 99.

Example:

```
Route map office>entry 1 match ip route-source 1
Route map office>
```

2.2.3.8 [no] entry <n> match ipv6 address

This command determines the IPv6 access list to be used when finding out if an IPv6 packet or an IPv6 route matches the entry.

In Route Redistribution and in BGP route exchange, this command checks the destination network for the routes with the configured access list.

The access list associated through this command must have been previously created.

If the packet or route does not match the access list classified as *permit*, said packet or route is considered as not matching the route map entry and the next entry on the route map is checked.

Syntax:

```
Route map <map-id>>entry <n> match ipv6 address <access-list>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

access-list Name of the IPv6 access control list.

Example:

```
Route map office>entry 1 match ipv6 address list1
Route map office>
```

2.2.3.9 [no] entry <n> match ipv6 next-hop

This command determines the access list to be used when finding out if the next hop for a route matches the entry.

This clause is only applied in Route Redistribution and in BGP route exchange.

The associated access list must have been previously created.

If the route does not match any access list classified as *permit*, then said route is considered not matching the route map entry and the next entry on the route map is checked.

Syntax:

```
Route map <map-id>>entry <n> match ipv6 nexthop <access-list>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

access-list Name of the IPv6 access control list.

Example:

```
Route map office>entry 1 match ipv6 next-hop list1
Route map office>
```

2.2.3.10 [no] entry <n> match length

This command establishes the maximum and minimum lengths of a packet in order to match the entry.

This clause is only applicable in Policy Routing.

The length considered is level 3 i.e. the total length of the IP packet (header plus data). This length is given in an IP header field.

Syntax:

```
Route map <map-id>>entry <n> match length <minimum-length> <maximum-length>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

minimum-length If the packet is shorter than this length, then it will not match the entry. The range goes from 0 to 4294967295.

maximum-length If the packet is longer than this length, then it will not match the entry. The range goes from 0 to 4294967295.

Example:

```
Route map office>entry 1 match length 0 127
Route map office>
```

2.2.3.11 [no] entry <n> match community

This command establishes the permitted communities so a BGP route matches an entry.

This clause is only applicable in BGP.

Syntax:

```
Route map <map-id>>entry <n> match community <community>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

community Defines one of the communities that match the entry. You can specify this in numeric format <AA:NN> or with a predefined community name (well-known). noadvertise, noexport or noexportsubconfed.

The predefined names for the community field are given in the following table:

Number	Name
65535:65281	no-export
65535:65282	no-advertise
65535:65283	no-export-subconfed

Example:

```
Route map office>entry 1 match community 100:35
Route map office>
```

2.2.3.12 [no] entry <n> match interface

This command establishes the permitted output interfaces so a route matches an entry.

This clause is only applied in route Redistribution and in OSPF route filtering.

Syntax:

```
Route map <map-id>>entry <n> match interface <interface>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

interface One of the interfaces present in the device.

Example:

```
Route map office>entry 1 match interface eth0/0
Route map office>
```

2.2.3.13 [no] entry <n> match metric

This command establishes the permitted metric so a route matches the entry.

This clause is only applied in route Redistribution and in OSPF route filtering.

Syntax:

```
Route map <map-id>>entry <n> match metric <metric>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

metric Metric value: The range is from 0 to 2147483647.

Example:

```
Route map office>entry 1 match metric 2
Route map office>
```

2.2.3.14 [no] entry <n> match origin

Adds filtering clauses, the checking of the “origin” attribute. This clause is only applicable in BGP routes (it is ignored for the rest of the routes).

Syntax:

```
Route map <id-mapa>>entry <n> match origin <origen>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

origen IGP. Routes originating in the initial Autonomous System.

EGP. Routes learned through the EGP protocol.

INCOMPLETE. Routes whose origins are unknown.

2.2.3.15 [no] entry <n> match route-type

This command establishes the type of permitted route so a route matches the entry.

This clause is only applied in route Redistribution and in OSPF route filtering.

Syntax:

```
Route map <map-id>>entry <n> match match route-type <typo>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

type The following selection of types of OSPF routes are supported:

internal. Intra-area and inter-area OSPF internal routes.

external. Type 1 and type 2 OSPF external routes.

external-type-1. OSPF type 1 external routes.

external-type-2. OSPF type 2 external routes

Example:

```
Route map office>entry 1 match route-type external
Route map office>
```

2.2.3.16 [no] entry <n> match tag

This command establishes the OSPF external route tags permitted so a route matches the entry.

This clause is only applied in OSPF route filtering.

Syntax:

```
Route map <map-id>>entry <n> match match tag <tag>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

tag Tag associated to an OSPF external route. This can take values between 0 and 4294967295.

Example:

```
Route map office>entry 1 match tag 0
Route map office>
```

2.2.3.17 [no] entry <n> set app-id

This command sets a custom AppId to an Advanced Forwarding System (AFS) session. Custom AppId has the format of a USER-defined (6) classification engine, followed by the custom application ID. The AppId is a parameter of AFS sessions that can be used as a selection criteria in stateful access lists, exportable through Netflow/IPFIX.

This clause is only applicable to Policy Routing if AFS is enabled.

The AppId of the packets matching the entry will be set to the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set app-id custom id <1..65535>
```

Example:

```
Route map office>entry 1 set app-id custom id 10
Route map office>
```

Command history:

Version	Modification
11.1.1	This command was introduced as of version 11.1.2

2.2.3.18 [no] entry <n> set as-path prepend

This command allows you to modify the AS-PATH attribute for a BGP route. This, in turn, influences the BGP decision process by putting the specified AS numbers before the route path. This clause is only applied when a BGP route is exported to an external autonomous system.

Syntax:

```
Route map <id-mapa>>entry <n> set as-path prepend <as-number>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

as-number AS number. The range goes from 1 to 65535 and can be repeated.

Example:

```
Route map office>entry 1 set as-path prepend 101
Route map office>entry 1 set as-path prepend 102
Route map office>entry 1 set as-path prepend 103
Route map office>
```

2.2.3.19 [no] entry <n> set ip default next-hop

This command determines the IP address or interface to be used when routing a packet if an adequate specified route cannot be found. It also allows you to specify if the packet is to be locally processed.

This clause is only applicable to Policy Routing, and in BGP.

When a specified route cannot be found to route a packet (a route that isn't a default route), this resorts to the IP addresses and interfaces configured through this command and takes the first IP address or interface that provides an active output interface. Moreover, the packet is locally processed if it first reaches a *local* clause.

When this is applied in BGP, the command modifies the next hop of the imported/exported route.

Syntax:

```
Route map <map-id>>entry <n> set ip default next-hop {<ip-address> | <interface> |
local | peer-address } [...<ip-address> | ...<interface> | ...local]
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from from 1 to 65535.

ip-address | interface | local | peer-address IP address for the next hop or output interface. If local is specified, this means that the packet must be locally processed instead of being routed. If the peer-address is specified (this is only used in BGP), the corresponding BGP peer address is used as the next hop.

Example:

```
Route map office>entry 1 set ip default next-hop 192.168.0.1
Route map office>
```

2.2.3.20 [no] entry <n> set ip df

This command causes a certain value to be assigned to the IP header DF bit for the packets which match this entry.

This clause is only applied in Policy Routing.

All packets that match the entry have the DF bit set with the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip df <bit-df>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

bit-df Value to establish in the IP header DF bit: 0 or 1.

Example:

```
Route map office>entry 1 set ip df 1
Route map office>
```

2.2.3.21 [no] entry <n> set ip next-hop

This command determines the IP address or interface to be used when routing packets that match the entry. This also allows you to specify if the packet is to be locally processed.

This clause is only applicable to Policy Routing, and in BGP.

When a packet matches the entry, this resorts to the IP addresses and interfaces configured through this command and routes the packet towards the first IP address or interface that provides an active output interface, or the packet is locally processed if it first reaches a *local* clause.

When applied in BGP, this modifies the next hop for the imported/exported route.

Syntax:

```
Route map <map-id>>entry <n> set ip next-hop {<ip-address> | <interface> | local | peer-address} [...<ip-address> | ...<interfa
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

ip-address | interface | local | peer-address IP address for the next hop or output interface. If local is specified, this means that the packet must be locally processed instead of being routed. If the peer-address is specified (this is only used in BGP), the corresponding BGP peer address is used as the next hop.

Example:

```
Route map office>entry 1 set ip next-hop 192.168.0.1
Route map office>
```

2.2.3.22 [no] entry <n> set ip precedence

This command establishes the IP header precedence field for packets matching the entry.

This clause is only applicable to Policy Routing.

All packets matching the entry will have the precedence field established with the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip precedence <precedence>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

precedence Value to establish in the IP header precedence field. This can be specified in numeric format or through a predefined name. The range goes from 0 to 7.

The predefined names for the precedence field are given in the following table:

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical

6	internet
7	network

Example:

```
Route map office>entry 1 set ip precedence priority
Route map office>
```

2.2.3.23 [no] entry <n> set ip tos

This command establishes the IP header tos field for packets matching the entry.

This clause is only applicable to Policy Routing.

The tos field in all packets matching the entry will be set to the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip tos <type-of-service>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

type-of-service Value to establish in the IP header tos field. This can be specified in numeric format or through a predefined name. The range goes from 0 to 15.

The predefined names for the type-of-service field are given in the following table:

Number	Name
0	normal
1	min-monetary-cost
2	max-reliability
4	max-throughput
8	min-delay

Example:

```
Route map office>entry 1 set ip tos min-delay
Route map office>
```

2.2.3.24 [no] entry <n> set ip dscp

This command establishes the IP header dscp field for packets matching the entry.

This clause is only applicable to Policy Routing.

The dscp field in all packets matching the entry will be set to the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip dscp <dscp>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

dscp Value to establish in the IP header dscp field. This must be specified in numeric format. The range goes from 0 to 63.

Example:

```
Route map office>entry 1 set ip dscp 15
Route map office>
```

2.2.3.25 [no] entry <n> set ip tos-octet

This command establishes the IP header tos byte for packets matching the entry.

This clause is only applicable to Policy Routing.

The tos byte in all packets matching the entry will be set to the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip tos-octet <byte-tos> [mask <mask>]
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.
byte-tos	Value to establish in the IP header tos byte. This must be specified in numeric format. The range goes from 0 to 255.
mask	Value of the bits mask which determines the bits to be marked. The range goes from 1 to 255. If this is not specified, a value of 255 is taken (all the byte bits are used).

Example:

```
Route map office>entry 1 set ip tos-octet 184
Route map office>
```

2.2.3.26 [no] entry <n> set ip session-mark

This command sets a session mark to an Advanced Forwarding System (AFS) session. This is an internal parameter, associated to the session, used as a selection criteria in an access-list and exported through Netflow/IPFIX.

This clause is only applicable to Policy Routing if AFS is enabled.

The session mark in all packets matching the entry will be set to the value configured through this command.

Syntax:

```
Route map <map-id>>entry <n> set ip session-mark <0..65535>
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.
session-mark	Value to establish in the session. It must be specified in numeric format. The range goes from 0 to 65535.

Example:

```
Route map office>entry 1 set ip session-mark 100
Route map office>
```

2.2.3.27 [no] entry <n> set ipv6 next-hop

This clause is only applied in BGP and in Route Redistribution.

When a packet matches the entry, the route's next hop is modified.

Syntax:

```
Route map <map-id>>entry <n> set ipv6 next-hop {<ipv6-address> | peer-address}
[...<ipv6-address>]
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.
ipv6-address | peer-address	IPv6 address for the next hop. If the peer-address is specified (this is only used in BGP), the corresponding BGP peer address is used as the next hop.

Example:

```
Route map office>entry 1 set ipv6 next-hop 2001:db8::1
Route map office>
```

2.2.3.28 [no] entry <n> set community

This command adds a community to the routes matching an entry.

This clause is only applicable in BGP.

All the routes matching the entry will have the communities configured through this command added to them.

Syntax:

```
Route map <map-id>>entry <n> set community <community>
```

n	Entry number. Despite entries not requiring consecutive numbers, the numerical order is import-

ant as the first number checked is the lowest. Values range from 1 to 65535.

community Community to add in the route attributes. You can specify this in numeric format <AA:NN> or with a predefined community name (well-known). noadvertise, noexport or noexportsubconfed.

The predefined names for the community field are given in the following table:

Number	Name
65535:65281	no-export
65535:65282	no-advertise
65535:65283	no-export-subconfed

Example:

```
Route map office>entry 1 set community 100:35
Route map office>
```

2.2.3.29 [no] entry <n> set label

This command modifies the label value for the IP packets that match the entry. The label is an internal parameter, associated to each IP packet, and consists of a number between 0 and 99. The label value for all IP packets is 0 by default. This attribute can be subsequently used to filter and classify the packet, using the access lists.

Syntax:

```
Route map <map-id>>entry <n> set label <value>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

label New label value which is associated to the IP packets that match the entry. The range goes between 0 and 99.

Example:

```
Route map office>entry 1 set label 26
Route map office>
```

2.2.3.30 [no] entry <n> set local-preference

This command establishes the Local Preference attribute in the routes matching the entry.

This clause is only applicable to BGP.

All routes matching the entry will have the Local Preference attribute, configured through this command, assigned to them.

Syntax:

```
Route map <map-id>>entry <n> set local-preference <preference>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

preference Preference value to establish in the Local Preference attribute for the route. The range goes from 0 to 255.

Example:

```
Route map office>entry 1 set local-preference 90
Route map office>
```

2.2.3.31 [no] entry <n> set metric

This command establishes the metric or cost for the routes matching the entry.

This clause is only applicable to route Redistribution and in BGP.

All routes matching the entry will have the metric or cost configured through this command assigned to them.

Syntax:

```
Route map <map-id>>entry <n> set metric <metric>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

metric Metric or cost to establish when redistributing the route.

Example:

```
Route map office>entry 1 set metric 5
Route map office>
```

2.2.3.32 [no] entry <n> set metric-type

This command establishes the type of OSPF route for the routes matching the entry.

This clause is only applicable to routes being redistributed to the OSPF protocol.

All routes matching the entry will have the type of OSPF external route configured through this command assigned to them.

Syntax:

```
Route map <map-id>>entry <n> set metric-type {type-1 | type-2}
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

type-1 Redistributes the route as a type 1 OSPF external route.

type-2 Redistributes the route as a type 2 OSPF external route.

Example:

```
Route map office>entry 1 set metric-type type-1
Route map office>
```

2.2.3.33 [no] entry <n> set origin

Allows you to modify the “origin” attribute for the routes that verify the checking clauses. As this is a BGP attribute, it is only applicable to the routes in this protocol.

Syntax:

```
Route map <id-mapa>>entry <n> set origin <origen>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

origen IGP. Routes originating in the initial Autonomous System.

EGP. Routes learned through the EGP protocol.

INCOMPLETE. Routes whose origins are unknown.

2.2.3.34 [no] entry <n> track nsla-advisor

This command links the entry to an external entity, specifically to an NSLA advisor. The entry is only applied if the linked NSLA advisor is activated (TRUE). In cases where the linked advisor is deactivated (FALSE), this entry is ignored (just as if it didn’t match).

Syntax:

```
Route map <map-id>>entry <n> track nsla-advisor <advisor-id>
```

n Entry number. Despite entries not requiring consecutive numbers, the numerical order is important as the first number checked is the lowest. Values range from 1 to 65535.

advisor-id Identifier for the NSLA advisor linked to the entry. The range is from 1 to 65535.

Example:

```
Route map office>entry 1 track nsla-advisor 1
Route map office>
```

2.2.3.35 no entry <n>

This command eliminates the specified entry. No other entry is affected.

Syntax:

```
Route map <map-id>>no entry <n>
```

n Entry number to eliminate. The range goes from 1 to 65535

Example:

```
Route map office>no entry 1
Route map office>
```

2.2.4 exit

This command finalizes configuration for a route-map and returns to the route-maps configuration menu.

Syntax:

```
Route map <id-map>>exit
```

Example:

```
Route map office>exit
Route map config>
```

Chapter 3 Monitoring Route maps

3.1 Monitoring Tools

Route-maps are only a routing policy database. Therefore, they do not have specific monitoring commands. To monitor the use of route-maps, use the monitoring tools available for the functionality applied (Policy routing, for example).