



BGP Support for IP Prefix Export from a VRF Table into the Global Table

This feature allows a network administrator to export IP prefixes from a VRF table into the global routing table.

- [Finding Feature Information, on page 1](#)
- [Information About IP Prefix Export from a VRF Table into the Global Table, on page 1](#)
- [How to Export IP Prefixes from a VRF Table into the Global Table, on page 3](#)
- [Configuration Examples for IP Prefix Export from a VRF Table into the Global Table, on page 9](#)
- [Additional References, on page 10](#)
- [Feature Information for IP Prefix Export from a VRF Table into the Global Table, on page 10](#)

Finding Feature Information

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

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

Information About IP Prefix Export from a VRF Table into the Global Table

Benefits of IP Prefix Export from a VRF Table into the Global Table

- You can manage some network resources inside a VRF by using a network management node residing in the global table.
- You own some internet public IP address space, but prefer to have a VRF to manage those IP addresses.

How IP Prefix Export from a VRF Table into the Global Table Works

MPLS-VPN using Multiprotocol BGP (MP-BGP) provides a very flexible but secured VPN provisioning mechanism for service providers and customers. However, some customers prefer to relax the boundary so that some specific prefixes can be reachable in a VRF as well as in the global routing table.

Prior to the BGP Support for IP Prefix Export from a VRF Table into Global Table feature, BGP already supported the global-to-VRF import of prefixes. See the “*BGP Support for IP Prefix Import from Global Table into a VRF Table*” module for complete documentation of that feature. Together, the import feature and export feature provide L3VPN dynamic route leaking.

The BGP Support for IP Prefix Export from a VRF Table into the Global Table feature provides the reverse mechanism of the import feature referenced above; it supports the export of prefixes from a VRF table to the global routing table. It is achieved with an **export {ipv4 | ipv6} {unicast | multicast} map** command, which specifies a route map to control the prefixes that are exported from a VRF table to the global routing table.



Caution

The IP Prefix Export from a VRF Table into Global Table feature leaks VRF routes into the global BGP routing table; those routes will be installed into the IPv4 or IPv6 routing table. Use extreme caution to design the network so that such leaking does not affect the normal Internet routing.

Export actions are triggered when a new routing update is received or when routes are withdrawn. During the initial BGP update period, the export action is postponed to allow BGP to converge more quickly. Once BGP converges, incremental BGP updates are evaluated immediately and qualified prefixes are exported as they are received.

Each VRF can export to only one of the global topologies in IPv4 (unicast or multicast) and can export to only one of the global topologies in IPv6 (unicast or multicast).

There is no limit to the number of VRFs per router that can be configured to export IPv4 or IPv6 prefixes to the global routing table.

By default, the software limits the number of prefixes that can be exported per VRF to 1000 prefixes. You can change that limit to a number in the range from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you increase the prefix limit above 1000. Configuring the device to export too many prefixes can interrupt normal router operation.

The following **match** and **set** commands are supported in this feature:

- **match as-path**
- **match community [exact-match]**
- **match extcommunity**
- **match ip address [prefix-list]**
- **match ip next-hop**
- **match ip route-source**
- **match ipv6 address [prefix-list]**
- **match ipv6 route-source**
- **match ipv6 next-hop**

- **match policy-list**
- **match route-type**
- **set as-path prepend** [last-as]
- **set community additive**
- **set extcommunity** [cost | rt]
- **set extcomm-list delete**
- **set ip next-hop**
- **set ipv6 next-hop**
- **set local-preference**
- **set metric**
- **set origin**
- **set weight**



Note The **set ip vrf next-hop** and **set ipv6 vrf next-hop** commands are not supported in this feature.

How to Export IP Prefixes from a VRF Table into the Global Table

Creating the VRF and the Export Route Map for an Address Family

The IP prefixes that are defined for export are processed through a match clause in a route map. IP prefixes that pass through the route map are exported into the global routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {*ipv4* | *ipv6*}
6. **export** {*ipv4* | *ipv6*} {**unicast** | **multicast**} [*prefix-limit*] **map** *map-name*
7. **route-target import** *route-target-ext-community*
8. **route-target export** *route-target-ext-community*
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*]} | **prefix-list** *prefix-list-name* [*prefix-list-name*]

13. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Device(config)# vrf definition vpn1	Creates a VRF routing table and specifies the VRF name (or tag).
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:100	Creates routing and forwarding tables for the VRF instance. <ul style="list-style-type: none"> • There are two formats for configuring the argument. It can be configured in the <i>as-number:network number (ASN:nn)</i> format, as shown in the example, or it can be configured in the <i>IP address:network number (IP-address:nn)</i> format.
Step 5	address-family { <i>ipv4</i> <i>ipv6</i> } Example: Device(config-vrf)# address-family ipv4	Configures the IPv4 or IPv6 address family.
Step 6	export { <i>ipv4</i> <i>ipv6</i> } { <i>unicast</i> <i>multicast</i> } [<i>prefix-limit</i>] map <i>map-name</i> Example: Device(config-vrf-af)# export ipv4 unicast 500 map UNICAST	Exports IPv4 or IPv6 prefixes from the VRF table to the global routing table, filtered by the specified route map. <ul style="list-style-type: none"> • Specify ipv4 or ipv6, which you specified in Step 5. This example exports IPv4 unicast prefixes. • Based on this example, no more than 500 prefixes will be exported. • The prefixes exported are those that pass the route map.
Step 7	route-target import <i>route-target-ext-community</i> Example: Device(config-vrf-af)# route-target import 100:100	Creates a route-target extended community for a VRF instance. <ul style="list-style-type: none"> • For information about route-target import or export, see the <i>MPLS: Layer 3 VPNs Configuration Guide</i>.

	Command or Action	Purpose
Step 8	route-target export <i>route-target-ext-community</i> Example: <pre>Device(config-vrf-af)# route-target export 100:100</pre>	Creates a route-target extended community for a VRF instance.
Step 9	exit Example: <pre>Device(config-vrf-af)# exit</pre>	Exits address family configuration mode and enters global configuration mode.
Step 10	exit Example: <pre>Device(config-vrf)# exit</pre>	Exits VRF configuration mode and enters global configuration mode.
Step 11	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: <pre>Device(config)# route-map UNICAST permit 10</pre>	Enables policy routing. <ul style="list-style-type: none"> • The example creates a route map named UNICAST.
Step 12	match ip address { <i>acl-number</i> [<i>acl-number</i> <i>acl-name</i>] <i>acl-name</i> [<i>acl-name</i> <i>acl-number</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name</i>]} Example: <pre>Device(config-route-map)# match ip address 50</pre>	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. <ul style="list-style-type: none"> • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria. • Define the access list (not shown in this task); for example, access-list 50 permit 192.168.1.0 255.255.255.0.
Step 13	end Example: <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and returns to privileged EXEC mode.

Creating the VRF and the Export Route Map for a VRF (IPv4 only)

The IP prefixes that are defined for export are processed through a match clause in a route map. IP prefixes that pass through the route map are exported into the global routing table.

**Note**

- Only IPv4 unicast and multicast prefixes can be exported from a VRF table to the global routing table under the **ip vrf** command, as shown in this task. To export IPv6 prefixes, you must do so under the IPv6 address family; see the section “Creating the VRF and the Export Route Map Per Address Family.”
- IPv4 prefixes exported into the global routing table using this feature cannot be exported into a VPNv4 VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **export ipv4** {unicast | multicast} [*prefix-limit*] **map** *map-tag*
6. **route-target import** *route-target-ext-community*
7. **route-target export** *route-target-ext-community*
8. **exit**
9. **route-map** *map-tag* [permit | deny] [*sequence-number*]
10. **match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i> Example: Device(config)# ip vrf GREEN	Creates a VRF routing table and specifies the VRF name (or tag). <ul style="list-style-type: none"> • The ip vrf <i>vrf-name</i> command creates a VRF routing table and a CEF table, and both are named using the <i>vrf-name</i> argument. Associated with these tables is the default route distinguisher value.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:10	Creates routing and forwarding tables for the VRF instance. <ul style="list-style-type: none"> • There are two formats for configuring the argument. It can be configured in the <i>as-number:network number</i> (<i>ASN:nn</i>) format, as shown in the example, or it can

	Command or Action	Purpose
		be configured in the <i>IP-address:network number</i> format (<i>IP-address:nn</i>).
Step 5	export ipv4 {unicast multicast} [<i>prefix-limit</i>] map <i>map-tag</i> Example: <pre>Device(config-vrf)# export ipv4 unicast 500 map UNICAST</pre>	Exports IPv4 prefixes from the VRF table to the global routing table, filtered by the specified route map. <ul style="list-style-type: none"> • Unicast or multicast prefixes are specified. • By default, up to 1000 prefixes can be exported. The <i>prefix-limit</i> argument is used to specify a limit from 1 to 2,147,483,647 prefixes. • The example creates an export map that will export up to 500 unicast prefixes that pass through the route map named UNICAST.
Step 6	route-target import <i>route-target-ext-community</i> Example: <pre>Device(config-vrf)# route-target import 100:100</pre>	Creates a route-target extended community for a VRF instance. <ul style="list-style-type: none"> • For information about route-target import or export, see the <i>MPLS: Layer 3 VPNs Configuration Guide</i>.
Step 7	route-target export <i>route-target-ext-community</i> Example: <pre>Device(config-vrf)# route-target export 100:100</pre>	Creates a route-target extended community for a VRF instance.
Step 8	exit Example: <pre>Device(config-vrf)# exit</pre>	Exits VRF configuration mode and enters global configuration mode.
Step 9	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: <pre>Device(config)# route-map UNICAST permit 10</pre>	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. <ul style="list-style-type: none"> • The route map name must match the route map specified in Step 5. • The example creates a route map named UNICAST.
Step 10	match ip address { <i>acl-number</i> [<i>acl-number</i> <i>acl-name</i>] <i>acl-name</i> [<i>acl-name</i> <i>acl-number</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name</i>]} Example: <pre>Device(config-route-map)# match ip address 50</pre>	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. <ul style="list-style-type: none"> • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria.

	Command or Action	Purpose
Step 11	end Example: <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and returns to privileged EXEC mode.

Displaying Information About IP Prefix Export from a VRF into the Global Table

Perform any of the steps in this task to see information about the prefixes exported from a VRF table into the global table.

SUMMARY STEPS

1. **enable**
2. **show ip bgp {ipv4 | ipv6} {unicast | multicast} [prefix]**
3. **debug ip bgp import event**
4. **debug ip bgp import update**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ip bgp {ipv4 ipv6} {unicast multicast} [prefix] Example: <pre>Device# show ip bgp ipv4 unicast 192.168.1.1</pre>	Displays information about the imported path from a VRF to the global table.
Step 3	debug ip bgp import event Example: <pre>Device# debug ip bgp import event</pre>	Displays messages related to IPv4 prefix import events.
Step 4	debug ip bgp import update Example: <pre>Device# debug ip bgp import update</pre>	Displays messages related to IPv4 prefix import updates.

Configuration Examples for IP Prefix Export from a VRF Table into the Global Table

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv6 Address Family

```
vrf definition X
  rd 100:100
  address-family ipv6
    export ipv6 unicast map OnlyNet2000
    route-target import 100:100
    route-target export 100:100
  !
  ipv6 prefix-list net2000 permit 2000::/16
  !
  route-map OnlyNet2000 permit 10
  match ipv6 address prefix-list net2000
```

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv4 Address Family

```
vrf definition X
  rd 100:100
  address-family ipv4
    export ipv4 unicast map OnlyNet200
    route-target import 100:100
    route-target export 100:100
  !
  ip prefix-list net200 permit 200.0.0.0/8
  !
  route-map OnlyNet200 permit 10
  match ip address prefix-list net200
```

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IP VRF (IPv4 Only)

```
ip vrf vrfname
  rd 100:100
  export ipv4 unicast map OnlyNet200
  route-target import 100:100
  route-target export 100:100
  !
  ip prefix-list net200 permit 200.0.0.0/8
  !
  route-map OnlyNet200 permit 10
  match ip address prefix-list net200
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS BGP Command Reference
Use of route-target import and export	<i>MPLS: Layer 3 VPNs Configuration Guide</i>

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 IP Prefix Export from a VRF Table into the Global Table

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

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

Table 1: Feature Information for BGP Support for IP Prefix Export from a VRF Table into the Global Table

Feature Name	Releases	Feature Information
BGP Support for IP Prefix Export from a VRF Table into the Global Table	15.3(1)T	<p>This feature allows a network administrator to export IP prefixes from a VRF routing table into the global routing table.</p> <p>The following command was introduced: export map (VRF table to global table).</p> <p>The following commands were modified: debug ip bgp import and show ip bgp.</p>

