THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALLFAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2011 Cisco Systems, Inc. All rights reserved.
CONTENTS

Cisco BGP Overview 1
Finding Feature Information 1
Prerequisites for Cisco BGP 1
Restrictions for Cisco BGP 1
Information About Cisco BGP 2
   BGP Version 4 Functional Overview 2
   BGP Autonomous Systems 3
   BGP Autonomous System Number Formats 4
      Asdot Only Autonomous System Number Formatting 4
      Asplain as Default Autonomous System Number Formatting 4
      Reserved and Private Autonomous System Numbers 5
Multiprotocol BGP 6
Benefits of Using Multiprotocol BGP Versus BGP 6
Multiprotocol BGP Extensions for IP Multicast 7
NLRI Configuration CLI 8
Cisco BGP Address Family Model 9
IPv4 Address Family 11
IPv6 Address Family 11
CLNS Address Family 11
VPNv4 Address Family 12
L2VPN Address Family 12
   BGP CLI Removal Considerations 13
Where to Go Next 14
Additional References 14
Feature Information for Cisco BGP Overview 16

Configuring a Basic BGP Network 21
Finding Feature Information 21
Prerequisites for Configuring a Basic BGP Network 21
Restrictions for Configuring a Basic BGP Network 21
Information About Configuring a Basic BGP Network 22

BGP Version 4 22
BGP Router ID 23
BGP-Speaker and Peer Relationships 23
BGP Autonomous System Number Formats 23
Cisco Implementation of 4-Byte Autonomous System Numbers 25
BGP Peer Session Establishment 26
Cisco Implementation of BGP Global and Address Family Configuration Commands 27
BGP Session Reset 28
BGP Route Aggregation 29
BGP Route Aggregation AS-SET Information Generation 29
Routing Policy Change Management 30
Conditional BGP Route Injection 31
BGP Peer Groups 31
BGP Backdoor Routes 31
BGP Update Group 32
BGP Dynamic Update Group Configuration 32
BGP Peer Templates 33
Inheritance in Peer Templates 33
Peer Session Templates 34
Peer Policy Templates 35
BGP IPv6 Neighbor Activation Under the IPv4 Address Family 36

How to Configure a Basic BGP Network 36

Configuring a BGP Routing Process 37

Troubleshooting Tips 40
Configuring a BGP Peer 40

Troubleshooting Tips 43
What to Do Next 43

Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers 43

Troubleshooting Tips 46
Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers 46
Configuring a BGP Peer for the IPv4 VRF Address Family 50

Troubleshooting Tips 54
Customizing a BGP Peer 54
## Contents

Removing BGP Configuration Commands Using a Redistribution 60  
Monitoring and Maintaining Basic BGP 62  
Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing 62  
Resetting and Displaying Basic BGP Information 65  
Aggregating Route Prefixes Using BGP 67  
Redistributing a Static Aggregate Route into BGP 67  
Configuring Conditional Aggregate Routes Using BGP 69  
Suppressing and Unsuppressing Advertising Aggregated Routes Using BGP 70  
Suppressing Inactive Route Advertisement Using BGP 72  
Conditionally Advertising BGP Routes 74  
Originating BGP Routes 77  
Advertising a Default Route Using BGP 77  
Troubleshooting Tips 79  
Conditionally Injecting BGP Routes 79  
Troubleshooting Tips 83  
Originating BGP Routes Using Backdoor Routes 83  
Configuring a BGP Peer Group 85  
Configuring Peer Session Templates 87  
Configuring a Basic Peer Session Template 87  
What to Do Next 89  
Configuring Peer Session Template Inheritance with the `inherit peer-session` Command 90  
What to Do Next 92  
Configuring Peer Session Template Inheritance with the `neighbor inherit peer-session` Command 92  
What to Do Next 93  
Configuring Peer Policy Templates 93  
Configuring Basic Peer Policy Templates 94  
What to Do Next 95  
Configuring Peer Policy Template Inheritance with the `inherit peer-policy` Command 96  
Configuring Peer Policy Template Inheritance with the `neighbor inherit peer-policy` Command 98  
Monitoring and Maintaining BGP Dynamic Update Groups 100  
Troubleshooting Tips 101  
Configuration Examples for a Basic BGP Network 102  
Example Configuring a BGP Process and Customizing Peers 102
Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example 234
Influencing Outbound Path Selection Examples 236
Filtering BGP Prefixes with Prefix Lists Examples 237
  Filtering BGP Prefixes Using a Single Prefix List 237
  Filtering BGP Prefixes Using a Group of Prefixes 238
  Adding or Deleting Prefix List Entries 238
Filtering Traffic Using Community Lists Examples 238
Filtering Traffic Using AS-path Filters Example 239
Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers Examples 240
Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example 240
Filtering Traffic Using a BGP Route Map Example 243
Filtering Traffic Using Continue Clauses in a BGP Route Map Example 243
Where to Go Next 244
Additional References 245
Feature Information for Connecting to a Service Provider Using External BGP 246

Configuring BGP Neighbor Session Options 253
  Finding Feature Information 253
  Prerequisites for Configuring BGP Neighbor Session Options 253
  Restrictions for Configuring BGP Neighbor Session Options 254
  Information About Configuring BGP Neighbor Session Options 254
    BGP Neighbor Sessions 254
    BGP Support for Fast Peering Session Deactivation 254
      BGP Hold Timer 254
      BGP Fast Peering Session Deactivation 254
    Selective Address Tracking for BGP Fast Session Deactivation 255
    BFD Support of BGP IPv6 Neighbors 255
    BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached 255
      Prefix Limits and BGP Peering Sessions 255
      BGP Neighbor Session Restart with the Maximum Prefix Limit 255
    BGP Network Autonomous System Migration 256
      Autonomous System Migration for BGP Networks 256
      Dual Autonomous System Support for BGP Network Autonomous System Migration 256
    TTL Security Check for BGP Neighbor Sessions 257
Examples Configuring BGP Support for TCP Path MTU Discovery per Session 295
Example Disabling TCP Path MTU Discovery Globally for All BGP Sessions 295
Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor 295
Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions 296
Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor 296
Example Implementing BGP Dynamic Neighbors Using Subnet Ranges 296

Where to Go Next 298
Additional References 298
Feature Information for Configuring BGP Neighbor Session Options 300

**Configuring Internal BGP Features** 305

Finding Feature Information 305
Information About Internal BGP Features 305
BGP Routing Domain Confederation 305
BGP Route Reflector 306
Route Reflector Mechanisms to Avoid Routing Loops 309
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer 309
BGP VPLS Autodiscovery Support on Route Reflector 310
BGP Route Dampening 310
Route Dampening Minimizes Route Flapping 310
BGP Route Dampening Terms 310

How To Configure Internal BGP Features 311
Configuring a Routing Domain Confederation 311
Configuring a Route Reflector 312
Configuring a Route Reflector Using a Route Map to Set Next Hop for iBGP Peer 312
Adjusting BGP Timers 316
Configuring the Router to Consider a Missing MED as Worst Path 317
Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths 317
Configuring the Router to Use the MED to Choose a Path in a Confederation 317
Enabling BGP Route Dampening 318
Monitoring and Maintaining BGP Route Dampening 318

Internal BGP Feature Configuration Examples 320
Example BGP Confederation Configurations with Route Maps 320
Examples BGP Confederation 320
Example Route Reflector Using a Route Map to Set Next Hop for iBGP Peer 321
Example BGP VPLS Autodiscovery Support on Route Reflector 322
Additional References 322
Feature Information for Configuring Internal BGP Features 324

Configuring Advanced BGP Features 327
Finding Feature Information 327
Prerequisites for Configuring Advanced BGP Features 327
Restrictions for Configuring Advanced BGP Features 327
Information About Configuring Advanced BGP Features 328

BGP Version 4 328
BGP Support for Next-Hop Address Tracking 328
BGP Next-Hop Address Tracking 328
Default BGP Scanner Behavior 329
Selective BGP Next-Hop Route Filtering 329
BGP Next_Hop Attribute 329
BGP Nonstop Forwarding Awareness 329
Cisco NSF Routing and Forwarding Operation 330
Cisco Express Forwarding for NSF 330
BGP Graceful Restart for NSF 331
BGP NSF Awareness 331
BGP Graceful Restart per Neighbor 332
BGP Peer Session Templates 332
BGP Route Dampening 333
BFD for BGP 334
BGP MIB Support 334

How to Configure Advanced BGP Features 336
Configuring BGP Next-Hop Address Tracking 336
Disabling BGP Next-Hop Address Tracking 336
Adjusting the Delay Interval for BGP Next-Hop Address Tracking 338
Configuring BGP Selective Next-Hop Route Filtering 339
Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart 343
Enabling BGP Global NSF Awareness Using BGP Graceful Restart 343
Troubleshooting Tips 344
What to Do Next 345
Configuring BGP NSF Awareness Timers 345
What to Do Next 346
Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates 347
Enabling BGP Graceful Restart for an Individual BGP Neighbor 352
Disabling BGP Graceful Restart for a BGP Peer Group 354
Verifying the Configuration of BGP Nonstop Forwarding Awareness 357
Configuring BGP Route Dampening 358
Enabling and Configuring BGP Route Dampening 358
Monitoring and Maintaining BGP Route Dampening 359
Decreasing BGP Convergence Time Using BFD 361
Prerequisites 361
Restrictions 361
Configuring BFD Session Parameters on the Interface 362
Configuring BFD Support for BGP 363
What to Do Next 364
Enabling BGP MIB Support 364
Configuration Examples for Configuring Advanced BGP Features 365
Enabling and Disabling BGP Next-Hop Address Tracking Example 366
Adjusting the Delay Interval for BGP Next-Hop Address Tracking Example 366
Configuring BGP Selective Next-Hop Route Filtering Examples 366
Enabling BGP Global NSF Awareness Using Graceful Restart Example 366
Enabling and Disabling BGP Graceful Restart per Neighbor Examples 367
Configuring BGP Route Dampening Example 368
Enabling BGP MIB Support Examples 368
Where to Go Next 369
Additional References 369
Feature Information for Configuring Advanced BGP Features 370
BGP Link Bandwidth 377
Finding Feature Information 377
Prerequisites for BGP Link Bandwidth 377
Restrictions for BGP Link Bandwidth 378
Information About BGP Link Bandwidth 378
BGP Link Bandwidth Overview 378
Link Bandwidth Extended Community Attribute 378
Benefits of the BGP Link Bandwidth Feature 378
How to Configure BGP Link Bandwidth 379
Configuring BGP Link Bandwidth 379
Contents

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2

1. eBGP and iBGP Multipath Load Sharing Verification Examples 405
   - Where to Go Next 406
   - Additional References 406
   - Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 408

2. Loadsharing IP Packets over More Than Six Parallel Paths 409
   - Finding Feature Information 409
   - Overview of Loadsharing IP Packets over More Than Six Parallel Paths 409
   - Additional References 410
   - Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths 411

3. BGP Policy Accounting Output Interface Accounting 413
   - Finding Feature Information 413
   - Prerequisites for BGP PA Output Interface Accounting 413
   - Information About BGP PA Output Interface Accounting 413
     - BGP PA Output Interface Accounting 414
     - Benefits of BGP PA Output Interface Accounting 414
   - How to Configure BGP PA Output Interface Accounting 415
     - Specifying the Match Criteria for BGP PA 415
     - Classifying the IP Traffic and Enabling BGP PA 416
     - Verifying BGP Policy Accounting 419
   - Configuration Examples for BGP PA Output Interface Accounting 421
     - Specifying the Match Criteria for BGP Policy Accounting Example 421
     - Classifying the IP Traffic and Enabling BGP Policy Accounting Example 422
   - Additional References 422
   - Feature Information for BGP Policy Accounting Output Interface Accounting 423
   - Glossary 424

4. BGP Cost Community 427
   - Finding Feature Information 427
   - Prerequisites for the BGP Cost Community Feature 427
   - Restrictions for the BGP Cost Community Feature 427
   - Information About the BGP Cost Community Feature 428
     - BGP Cost Community Overview 428
     - How the BGP Cost Community Influences the Best Path Selection Process 428
     - Cost Community Support for Aggregate Routes and Multipaths 429
     - Influencing Route Preference in a Multi-Exit IGP Network 430
     - BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links 430

Additional References

Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 408
Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths 411
Feature Information for BGP Policy Accounting Output Interface Accounting 423
Feature Information for BGP Policy Accounting Output Interface Accounting 423
Glossary 424

Note: The page numbers are placeholders and indicate the general section and page number where the feature or topic is discussed.
Example Configuring BGP PIC  571
Example Displaying Backup Alternate Paths for BGP PIC  572
Additional References  574
Feature Information for BGP PIC  575
Cisco BGP Overview

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS XE software implementation of BGP version 4 includes support for 4-byte autonomous system numbers and multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), Connectionless Network Services (CLNS), and Layer 2 VPN (L2VPN). This module contains conceptual material to help you understand how BGP is implemented in Cisco IOS software.

- Finding Feature Information, page 1
- Prerequisites for Cisco BGP, page 1
- Restrictions for Cisco BGP, page 1
- Information About Cisco BGP, page 2
- Where to Go Next, page 14
- Additional References, page 14
- Feature Information for Cisco BGP Overview, page 16

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Cisco BGP

This document assumes knowledge of IPv4, IPv6, multicast, VPNv4, and Interior Gateway Protocols (IGPs). The amount of knowledge required for each technology is dependent on your deployment.

Restrictions for Cisco BGP

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple concurrent BGP address family and subaddress family configurations.
Information About Cisco BGP

- BGP Version 4 Functional Overview, page 2
- BGP Autonomous Systems, page 3
- BGP Autonomous System Number Formats, page 4
- Multiprotocol BGP, page 6
- Benefits of Using Multiprotocol BGP Versus BGP, page 6
- Multiprotocol BGP Extensions for IP Multicast, page 7
- NLRI Configuration CLI, page 8
- Cisco BGP Address Family Model, page 9
- IPv4 Address Family, page 11
- IPv6 Address Family, page 11
- CLNS Address Family, page 11
- VPNv4 Address Family, page 12
- L2VPN Address Family, page 12
- BGP CLI Removal Considerations, page 13

BGP Version 4 Functional Overview

BGP is an interdomain routing protocol that is designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol; it uses TCP (port 179) as the transport protocol because TCP is a connection-oriented protocol. The destination TCP port is assigned 179, and the local port is assigned a random port number. Cisco IOS XE software supports BGP version 4, and it is this version that has been used by Internet service providers to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use. RFC 2858 introduced multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IPv4 and IPv6.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. Although BGP is referred to as an Exterior Gateway Protocol (EGP), many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions. For more details about configuring BGP peer sessions and other tasks to build a basic BGP network, see the "Configuring a Basic BGP Network" module.

BGP uses a path-vector routing algorithm to exchange network reachability information with other BGP speaking networking devices. Network reachability information is exchanged between BGP peers in routing updates. Network reachability information contains the network number, path specific attributes, and the list of autonomous system numbers that a route must transit through to reach a destination network. This list is contained in the AS-path attribute. BGP prevents routing loops by rejecting any routing update that contains the local autonomous system number because this indicates that the route has already travelled through that autonomous system and a loop would therefore be created. The BGP path-vector routing algorithm is a combination of the distance-vector routing algorithm and the AS-path loop detection. For more details about configuration tasks to configure various options involving BGP neighbor peer sessions, see the "Configuring BGP Neighbor Session Options" module.

BGP selects a single path, by default, as the best path to a destination host or network. The best path selection algorithm analyzes path attributes to determine which route is installed as the best path in the
BGP routing table. Each path carries well-known mandatory, well-know discretionary, and optional transitive attributes that are used in BGP best path analysis. Cisco IOS XE software provides the ability to influence BGP path selection by altering some of these attributes using the command-line interface (CLI.) BGP path selection can also be influenced through standard BGP policy configuration. For more details about using BGP to influence path selection and configuring BGP policies to filter traffic, see the “Connecting to a Service Provider Using External BGP” module.

BGP can be used to help manage complex internal networks by interfacing with Interior Gateway Protocols (IGPs). Internal BGP can help with issues such as scaling the existing IGPs to match the traffic demands while maintaining network efficiency. For more details about configuring advanced BGP features including tasks to configure iBGP peering sessions, see the “Configuring Advanced BGP Features” module.

**BGP Autonomous Systems**

An autonomous system is a network controlled by a single technical administration entity. BGP autonomous systems are used to divide global external networks into individual routing domains where local routing policies are applied. This organization simplifies routing domain administration and simplifies consistent policy configuration. Consistent policy configuration is important to allow BGP to efficiently process routes to destination networks.

Each routing domain can support multiple routing protocols. However, each routing protocol is administrated separately. Other routing protocols can dynamically exchange routing information with BGP through redistribution. Separate BGP autonomous systems dynamically exchange routing information through eBGP peering sessions. BGP peers within the same autonomous system exchange routing information through iBGP peering sessions.

The figure below illustrates two routers in separate autonomous systems that can be connected using BGP. Router A and Router B are Internet service provider (ISP) routers in separate routing domains that use public autonomous system numbers. These routers carry traffic across the Internet. Router A and Router B are connected through eBGP peering sessions.

*Figure 1  BGP Topology with Two Autonomous Systems*

Each public autonomous system that directly connects to the Internet is assigned a unique number that identifies both the BGP routing process and the autonomous system.
BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were two-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority (IANA) will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- **Asplain--**Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- **Asdot--**Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

- [Asdot Only Autonomous System Number Formatting](#), page 4
- [Asplain as Default Autonomous System Number Formatting](#), page 4
- [Reserved and Private Autonomous System Numbers](#), page 5

### Asdot Only Autonomous System Number Formatting

In Cisco IOS XE Release 2.3 and later releases, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers, the asdot format includes a period, which is a special character in regular expressions. A backslash must be entered before the period; for example, `\1.14`, to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in `show` command output in Cisco IOS XE images where only asdot formatting is available.

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

### Asplain as Default Autonomous System Number Formatting

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you
want to change the default `show` command output to display 4-byte autonomous system numbers in the `asdot` format, use the `bgp asnotation dot` command in router configuration mode.

When the `asdot` format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the `asdot` format, or the regular expression match will fail. The tables below show that, although you can configure 4-byte autonomous system numbers in either `asplain` or `asdot` format, only one format is used to display `show` command output and control 4-byte autonomous system number matching for regular expressions, and the default is `asplain` format. To display 4-byte autonomous system numbers in `show` command output and to control matching for regular expressions in the `asdot` format, you must configure the `bgp asnotation dot` command. After you configure the `bgp asnotation dot` command, you must initiate a hard reset for all BGP sessions by entering the `clear ip bgp *` command.

**Note**

If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The `show` command output and regular expression match are not changed and remain in `asplain` (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Default Asplain 4-Byte Autonomous System Number Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Configuration Format</td>
</tr>
<tr>
<td>asplain</td>
<td>2-byte: 1 to 65535 4-byte: 65536 4-byte: 1.0 to 65535.65535</td>
</tr>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Asdot 4-Byte Autonomous System Number Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Configuration Format</td>
</tr>
<tr>
<td>asplain</td>
<td>2-byte: 1 to 65535 4-byte: 65536 4-byte: 1.0 to 65535.65535</td>
</tr>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

**Reserved and Private Autonomous System Numbers**

In Cisco IOS XE Release 2.3 and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.
RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allows configuration examples to be accurately documented and avoids conflict with production networks if these configurations are copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511, and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS XE software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.

**Note**

Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: [http://www.iana.org/](http://www.iana.org/).

**Multiprotocol BGP**

Cisco IOS XE software supports multiprotocol BGP extensions as defined in RFC 2858. The extensions introduced in this RFC allow BGP to carry routing information for multiple network-layer protocols including IPv4, IPv6, and VPNv4. These extensions are backward-compatible to enable routers that do not support multiprotocol extensions to communicate with those routers that do support multiprotocol extensions. Multiprotocol BGP carries routing information for multiple network-layer protocols and IP multicast routes. BGP carries different sets of routes depending on the protocol. For example, BGP can carry one set of routes for IPv4 unicast routing, one set of routes for IPv4 multicast routing, and one set of routes for MPLS VPNv4 routes.

**Note**

A multiprotocol BGP network is backward-compatible with a BGP network, but BGP peers that do not support multiprotocol extensions cannot forward routing information, such as address family identifier information, that the multiprotocol extensions carry.

**Benefits of Using Multiprotocol BGP Versus BGP**

In complex networks with multiple network layer protocols, multiprotocol BGP must be used. In less complex networks we recommend using multiprotocol BGP because it offers the following benefits:

- All of the BGP commands and routing policy capabilities of BGP can be applied to multiprotocol BGP.
- A network can carry routing information for multiple network layer protocol address families (for example, IP Version 4 or VPN Version 4) as specified in RFC 1700, *Assigned Numbers*.
- A network can support incongruent unicast and multicast topologies.
- A multiprotocol BGP network is backward compatible because the routers that support the multiprotocol extensions can interoperate with routers that do not support the extensions.
In summary, multiprotocol BGP support for multiple network layer protocol address families provides a flexible and scalable infrastructure that allows you to define independent policy and peering configurations on a per-address family basis.

**Multiprotocol BGP Extensions for IP Multicast**

The routes associated with multicast routing are used by the Protocol Independent Multicast (PIM) feature to build data distribution trees. Multiprotocol BGP is useful when you want a link dedicated to multicast traffic, perhaps to limit which resources are used for which traffic. For example, you want all multicast traffic exchanged at one network access point (NAP). Multiprotocol BGP allows you to have a unicast routing topology different from a multicast routing topology that allows you more control over your network and resources.

In BGP, the only way to perform interdomain multicast routing is to use the BGP infrastructure that is in place for unicast routing. If the routers are not multicast-capable, or there are differing policies about where multicast traffic should flow, multicast routing cannot be supported without multiprotocol BGP.

A multicast routing protocol, such as PIM, uses both the multicast and unicast BGP database to source the route, perform Reverse Path Forwarding (RPF) lookups for multicast-capable sources, and build a multicast distribution tree (MDT). The multicast table is the primary source for the router, but if the route is not found in the multicast table then the unicast table is searched. Although multicast can be performed with unicast BGP, multicast BGP routes allow an alternative topology to be used for RPF.

It is possible to configure BGP peers that exchange both unicast and multicast Network Layer Reachability Information (NLRI) where multiprotocol BGP routes can be redistributed into BGP. Multiprotocol extensions, however, will be ignored by any peers that do not support multiprotocol BGP. When PIM builds a multicast distribution tree through a unicast BGP network (because the route through the unicast network is the most attractive), the RPF check may fail, preventing the MDT from being built. If the unicast network runs multiprotocol BGP, peering can be configured using the appropriate multicast address family. The multicast address family configuration enables multiprotocol BGP to carry the multicast information and the RPF lookup will succeed.

The figure below illustrates a simple example of unicast and multicast topologies that are incongruent; these topologies cannot exchange information without implementing multiprotocol BGP. Autonomous systems 100, 200, and 300 are each connected to two NAPs that are FDDI rings. One is used for unicast peering (and therefore the exchanging of unicast traffic). The Multicast Friendly Interconnect (MFI) ring is used for multicast peering (and therefore the exchanging of multicast traffic). Each router is unicast- and multicast-capable.

**Figure 2 Incongruent Unicast and Multicast Routes**
The figure below is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they do not support or are not configured to perform multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both unicast and multicast routing. The unicast-only and multicast-only routers are connected to a single NAP.

In the figure below, only unicast traffic can travel from Router A to the unicast routers to Router B and back. Multicast traffic could not flow on that path, because multicast routing is not configured on the unicast routers and therefore the BGP routing table does not contain any multicast routes. On the multicast routers, multicast routes are enabled and BGP builds a separate routing table to hold the multicast routes. Multicast traffic uses the path from Router A to the multicast routers to Router B and back.

The figure below illustrates a multiprotocol BGP environment with a separate unicast route and multicast route from Router A to Router B. Multiprotocol BGP allows these routes to be noncongruent. Both of the autonomous systems must be configured for internal multiprotocol BGP in the figure.

**Figure 3  Multicast BGP Environment**

For more information about IP multicast, see the "Configuring IP Multicast" configuration library.

**NLRI Configuration CLI**

BGP was designed to carry only unicast IPv4 routing information. BGP configuration used the Network NLRI format CLI in Cisco IOS XE software. The NLRI format offers only limited support for multicast routing information and does not support multiple network layer protocols. We do not recommend using NLRI format CLI for BGP configuration.

Using the BGP hybrid CLI feature, you can configure commands in the address family VPNv4 format and save these command configurations without modifying an existing NLRI formatted configuration. If you
want to use other address family configurations such as IPv4 unicast or multicast, then you must upgrade the configuration using the `bgp upgrade-cli` command.

For more details about using BGP hybrid CLI command, see the "Configuring a Basic BGP Network" module. See the "Multiprotocol BGP" and "Cisco BGP Address Family Model" concepts for more information about address family configuration format and the limitations of the NLRI CLI format.

### Cisco BGP Address Family Model

The Cisco BGP address family identifier (AFI) model was introduced with multiprotocol BGP and is designed to be modular and scalable and to support multiple AFI and subsequent address family identifier (SAFI) configurations. Networks are increasing in complexity, and many companies are now using BGP to connect to many autonomous systems, as shown in the network topology in the figure below. Each of the separate autonomous systems shown in the figure below may be running several routing protocols such as Multiprotocol Label Switching (MPLS) and IPv6 and require both unicast and multicast routes to be transported via BGP.

**Figure 4  BGP Network Topology for Multiple Address Families**

![BGP Network Topology for Multiple Address Families](image)

Multiprotocol BGP carries routing information for multiple network layer protocols and IP multicast routes. This routing information is carried in the AFI model as appended BGP attributes (multiprotocol extensions). Each address family maintains a separate BGP database, which allows you to configure BGP policy on per-address family basis. SAFI configurations are subsets of the parent AFI. SAFIs can be used to refine BGP policy configurations.

The AFI model was created because of the scalability limitations of the NLRI format. A router that is configured in NLRI format has IPv4 unicast but limited multicast capabilities. Networks that are configured in the NLRI format have the following limitations:

- No support for AFI and SAFI configuration information. Many new BGP (and other protocols such as MPLS) features are supported only in AFI and SAFI configuration modes and cannot be configured in NLRI configuration modes.
- No support for IPv6. A router that is configured in the NLRI format cannot establish peering with an IPv6 neighbor.
- Limited support for multicast interdomain routing and incongruent multicast and unicast topologies. In the NLRI format, not all configuration options are available and there is no support for VPNv4. The
NLRI format configurations can be more complex than configurations that support the AFI model. If the routers in the infrastructure do not have multicast capabilities, or if policies differ as to where multicast traffic is configured to flow, multicast routing cannot be supported.

The AFI model in multiprotocol BGP supports multiple AFIs and SAFIs, all NLRI-based commands and policy configurations, and is backward-compatible with routers that support only the NLRI format. A router that is configured using the AFI model has the following features:

- AFI and SAFI information and configurations are supported. A router that is configured using the AFI model can carry routing information for multiple network layer protocol address families (for example, IPv4 and IPv6).
- AFI configuration is similar in all address families, making the CLI syntax easier to use than the NLRI format syntax.
- All BGP routing policy capabilities and commands are supported.
- Congruent unicast and multicast topologies that have different policies (BGP filtering configurations) are supported, as are incongruent multicast and unicast topologies.
- CLNS is supported.
- Interoperation between routers that support only the NLRI format (AFI-based networks are backward-compatible) is supported. This includes both IPv4 unicast and multicast NLRI peers.
- Virtual Private Networks (VPNs) and VPN routing and forwarding (VRF) instances are supported. Unicast IPv4 for VRFs can be configured from a specific address family IPv4 VRF; this configuration update is integrated into the BGP VPNv4 database.

Within a specific address family configuration mode, the question mark (?) online help function can be used to display supported commands. The BGP commands supported in address family configuration mode configure the same functionality as the BGP commands supported in router configuration mode; however, the BGP commands in router configuration mode configure functionality only for the IPv4 unicast address prefix. To configure BGP commands and functionality for other address family prefixes (for example, the IPv4 multicast or IPv6 unicast address prefixes), you must enter address family configuration mode for those address prefixes.

The BGP address family model consists of the following address families in Cisco IOS XE software: IPv4, IPv6, CLNS, and VPNv4. Within the L2VPN address family, the VPLS SAFI is supported. Within the IPv4 and IPv6 address families, SAFIs such as Multicast Distribution Tree (MDT), tunnel, and VRF exist. The table below shows the list of SAFIs supported by Cisco IOS XE software. To ensure compatibility between networks running all types of AFI and SAFI configuration, we recommend configuring BGP on Cisco IOS XE devices using the multiprotocol BGP address family model.

### Table 4 SAFIs Supported by Cisco IOS XE software

<table>
<thead>
<tr>
<th>SAFI Field Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NLRI used for unicast forwarding.</td>
<td>RFC 2858</td>
</tr>
<tr>
<td>2</td>
<td>NLRI used for multicast forwarding.</td>
<td>RFC 2858</td>
</tr>
<tr>
<td>3</td>
<td>NLRI used for both unicast and multicast forwarding.</td>
<td>RFC 2858</td>
</tr>
<tr>
<td>4</td>
<td>NLRI with MPLS labels.</td>
<td>RFC 3107</td>
</tr>
</tbody>
</table>
IPv4 Address Family

The IPv4 address family is used to identify routing sessions for protocols such as BGP that use standard IP version 4 address prefixes. Unicast or multicast address prefixes can be specified within the IPv4 address family. Routing information for address family IPv4 unicast is advertised by default when a BGP peer is configured unless the advertisement of unicast IPv4 information is explicitly turned off.

VRF instances can also be associated with IPv4 AFI configuration mode commands.

The tunnel SAFI was introduced to support multipoint tunneling IPv4 routing sessions. The tunnel SAFI is used to advertise the tunnel endpoints and the SAFI-specific attributes that contain the tunnel type and tunnel capabilities. Redistribution of tunnel endpoints into the BGP IPv4 tunnel SAFI table occurs automatically when the tunnel address family is configured. However, peers need to be activated under the tunnel address family before the sessions can exchange tunnel information.

The MDT SAFI was introduced to support multicast VPN architectures. The MDT SAFI is a transitive multicast-capable connector attribute that is defined as an IPv4 address family in BGP. The MDT address family session operates as a SAFI under the IPv4 multicast address family, and is configured on provider edge (PE) routers to establish VPN peering sessions with customer edge (CE) routers that support inter-AS multicast VPN peering sessions.

IPv6 Address Family

The IPv6 address family is used to identify routing sessions for protocols such as BGP that use standard IPv6 address prefixes. Unicast or multicast address prefixes can be specified within the IPv6 address family.

Note
Routing information for address family IPv4 unicast is advertised by default when you configure a BGP peer unless you explicitly turn off the advertisement of unicast IPv4 information.

CLNS Address Family

The CLNS address family is used to identify routing sessions for protocols such as BGP that use standard network service access point (NSAP) address prefixes. Unicast address prefixes are the default when NSAP address prefixes are configured.
CLNS routes are used in networks where CLNS addresses are configured. This is typically a telecommunications Data Communications Network (DCN). Peering is established using IP addresses, but update messages contain CLNS routes.

For more details about configuring BGP support for CLNS, which provides the ability to scale CLNS networks, see the "Configuring Multiprotocol BGP (MP-BGP) support for CLNS" module.

**VPNv4 Address Family**

The VPNv4 multicast address family is used to identify routing sessions for protocols such as BGP that use standard VPN Version 4 address prefixes. Unicast address prefixes are the default when VPNv4 address prefixes are configured. VPNv4 routes are the same as IPv4 routes, but VPNv4 routes have a route descriptor (RD) prepended that allows replication of prefixes. It is possible to associate every different RD with a different VPN. Each VPN needs its own set of prefixes.

Companies use an IP VPN as the foundation for deploying or administering value-added services including applications and data hosting network commerce, and telephony services to business customers. In private LANs, IP-based intranets have fundamentally changed the way companies conduct their business. Companies are moving their business applications to their intranets to extend over a WAN. Companies are also addressing the needs of their customers, suppliers, and partners by using extranets (an intranet that encompasses multiple businesses). With extranets, companies reduce business process costs by facilitating supply-chain automation, electronic data interchange (EDI), and other forms of network commerce. To take advantage of this business opportunity, service providers must have an IP VPN infrastructure that delivers private network services to businesses over a public infrastructure.

VPNs, when used with MPLS, allow several sites to transparently interconnect through a service provider’s network. One service provider network can support several different IP VPNs. Each of these appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN. Each VPN is associated with one or more VPN VRFs. VPNv4 routes are a superset of routes from all VRFs, and route injection is done per VRF under the specific VRF address family. The router maintains a separate routing and Cisco Express Forwarding table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems. The router using BGP distributes the VPN routing information using the BGP extended communities.

The VPN address space is isolated from the global address space by design. BGP distributes reachability information for VPN-IPv4 prefixes for each VPN using the VPNv4 multiprotocol extensions to ensure that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

RFC 3107 specifies how to add label information to multiprotocol BGP address families using a SAFI. The Cisco IOS XE implementation of MPLS uses RFC 3107 to provide support for sending IPv4 routes with a label. VPNv4 routes implicitly have a label associated with each route.

**L2VPN Address Family**

In Cisco IOS XE Release 2.6 and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or generic routing encapsulation (GRE). The L2VPN address family is configured in BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured.
Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the "VPLS Autodiscovery: BGP Based" feature.

Under L2VPN address family, the following BGP commands are supported:

- `bgp` nexthop
- `bgp` scan-time
- `neighbor` activate
- `neighbor` advertisement-interval
- `neighbor` allowas-in
- `neighbor` capability
- `neighbor` inherit
- `neighbor` peer-group
- `neighbor` maximum-prefix
- `neighbor` next-hop-self
- `neighbor` next-hop-unchanged
- `neighbor` remove-private-as
- `neighbor` route-map
- `neighbor` route-reflector-client
- `neighbor` soo
- `neighbor` weight

For route reflectors using L2VPNs, the `neighbor next-hop-self` and `neighbor next-hop-unchanged` commands are not supported.

For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration mode. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

For details on configuring BGP under the L2VPN address family, see the "BGP Support for the L2VPN Address Family" feature.

### BGP CLI Removal Considerations

BGP CLI configuration can become quite complex even in smaller BGP networks. If you need to remove any CLI configuration, you must consider all the implications of removing the CLI. Analyze the current running configuration to determine the current BGP neighbor relationships, any address family considerations, and even other routing protocols that are configured. Many BGP CLI commands affect other parts of the CLI configuration. For example, in the following configuration, a route map is used to...
match a BGP autonomous system number and then set the matched routes with another autonomous system number for EIGRP:

```plaintext
route-map bgp-to-eigrp permit 10
match tag 50000
set tag 65000
```

BGP neighbors in three different autonomous systems are configured and activated:

```plaintext
router bgp 45000
  bgp log-neighbor-changes
  address-family ipv4
    neighbor 172.16.1.2 remote-as 45000
    neighbor 192.168.1.2 remote-as 40000
    neighbor 192.168.3.2 remote-as 50000
    neighbor 172.16.1.2 activate
    neighbor 192.168.1.2 activate
    network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

An EIGRP routing process is then configured and BGP routes are redistributed into EIGRP with a route map filtering the routes:

```plaintext
router eigrp 100
  redistribute bgp 45000 metric 10000 100 255 1 1500 route-map bgp-to-eigrp
  no auto-summary
  exit
```

If you later decide to remove the route map, you will use the `no` form of the `route-map` command. Almost every configuration command has a `no` form, and the `no` form generally disables a function. However, in this configuration example, if you disable only the route map, the route redistribution will continue, but without the filtering or matching from the route map. Redistribution without the route map may cause unexpected results in your network. When you remove an access list or route map, you must also review the commands that referenced that access list or route map to consider whether the command will give you the behavior you intended.

The following configuration will remove both the route map and the redistribution:

```plaintext
configure terminal
  no route-map bgp-to-eigrp
  router eigrp 100
    no redistribute bgp 45000
  end
```

For details on configuring the removal of BGP CLI configuration, see the "Configuring a Basic BGP Network" module.

**Where to Go Next**

Proceed to the "Configuring a Basic BGP Network" module.

**Additional References**
### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Configuring basic BGP tasks</td>
<td>&quot;Configuring a Basic BGP Network&quot;</td>
</tr>
<tr>
<td>Configuring BGP neighbor session options</td>
<td>&quot;Configuring BGP Neighbor Session Options&quot;</td>
</tr>
<tr>
<td>Configuring BGP to connect to a service provider</td>
<td>&quot;Connecting to a Service Provider Using External BGP&quot;</td>
</tr>
<tr>
<td>Configuring internal BGP (iBGP) tasks</td>
<td>&quot;Configuring Internal BGP Features&quot;</td>
</tr>
<tr>
<td>Configuring advanced BGP features</td>
<td>&quot;Configuring Advanced BGP Features&quot;</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT SAFI</td>
<td>MDT SAFI</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1700</td>
<td>Assigned Numbers</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 3107</td>
<td>Carrying Label Information in BGP-4</td>
</tr>
<tr>
<td>RFC 4271</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 4893</td>
<td>BGP Support for Four-octet AS Number Space</td>
</tr>
<tr>
<td>RFC 5396</td>
<td>Textual Representation of Autonomous System (AS) Numbers</td>
</tr>
</tbody>
</table>
RFC 5398  Autonomous System (AS) Number Reservation for Documentation Use

Technical Assistance

Description
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.

To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.

Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.

Feature Information for Cisco BGP Overview

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>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP 4 Multipath Support</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Multicast BGP (MBGP)</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| BGP Support for 4-Byte ASN   | Cisco IOS XE Release 2.3 Cisco IOS XE Release 2.4 | The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295. In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte autonomous system numbers uses asdot as the only configuration format, regular expression match, and output display, with no asplain support. In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the `bgp asnotation dot` command. The following commands were introduced or modified by this feature: `bgp asnotation dot, bgp confederation identifier, bgp confederation peers, all clear ip bgp` commands that configure an autonomous system number, `ip as-path access-list, ip extcommunity-list, match source-protocol, neighbor local-as, neighbor remote-as, redistribute (IP), router bgp, route-target, set as-path, set`
BGP Support for the L2VPN Address Family

Cisco IOS XE Release 2.6  

BGP Support for the L2VPN address family introduced a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a Pseudowire mesh to support L2VPN-based services.

The following commands were introduced or modified by this feature:

- `address-family l2vpn`
- `show ip bgp l2vpn`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Configuring Multiprotocol BGP Support for CLNS | Cisco IOS XE Release 2.6 | The Multiprotocol BGP (MP-BGP) Support for CLNS feature provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks. The following commands were introduced or modified by this feature:  
  - clear bgp nsap  
  - clear bgp nsap dampening  
  - clear bgp nsap external  
  - clear bgp nsap flap-statistics  
  - clear bgp nsap peer-group  
  - debug bgp nsap  
  - debug bgp nsap dampening  
  - debug bgp nsap updates  
  - neighbor prefix-list  
  - network (BGP and multiprotocol BGP)  
  - redistribute (BGP to ISO ISIS)  
  - redistribute (ISO ISIS to BGP)  
  - show bgp nsap  
  - show bgp nsap community  
  - show bgp nsap community-list  
  - show bgp nsap dampened-paths  
  - show bgp nsap filter-list  
  - show bgp nsap flap-statistics  
  - show bgp nsap inconsistent-as  
  - show bgp nsap neighbors |
### Feature Name | Releases | Feature Information
--- | --- | ---
 |  | • show bgp nsap paths
 |  | • show bgp nsap quote-regexp
 |  | • show bgp nsap regexp
 |  | • show bgp nsap summary

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring a Basic BGP Network

This module describes the basic tasks to configure a basic Border Gateway Protocol (BGP) network. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. The Cisco IOS XE implementation of the neighbor and address family commands is explained. This module also contains tasks to configure and customize BGP peers and to configure BGP route aggregation, BGP route origination, BGP backdoor routes, BGP peer groups, peer session templates, and update groups.

• Finding Feature Information, page 21
• Prerequisites for Configuring a Basic BGP Network, page 21
• Restrictions for Configuring a Basic BGP Network, page 21
• Information About Configuring a Basic BGP Network, page 22
• How to Configure a Basic BGP Network, page 36
• Configuration Examples for a Basic BGP Network, page 102
• Where to Go Next, page 114
• Additional References, page 115
• Feature Information for Configuring a Basic BGP Network, page 116

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Configuring a Basic BGP Network

Before configuring a basic BGP network, you should be familiar with the "Cisco BGP Overview" module.

Restrictions for Configuring a Basic BGP Network

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.
Information About Configuring a Basic BGP Network

- BGP Version 4, page 22
- BGP Router ID, page 23
- BGP-Speaker and Peer Relationships, page 23
- BGP Autonomous System Number Formats, page 23
- Cisco Implementation of 4-Byte Autonomous System Numbers, page 25
- BGP Peer Session Establishment, page 26
- Cisco Implementation of BGP Global and Address Family Configuration Commands, page 27
- BGP Session Reset, page 28
- BGP Route Aggregation, page 29
- BGP Route Aggregation AS-SET Information Generation, page 29
- Routing Policy Change Management, page 30
- Conditional BGP Route Injection, page 31
- BGP Peer Groups, page 31
- BGP Backdoor Routes, page 31
- BGP Update Group, page 32
- BGP Dynamic Update Group Configuration, page 32
- BGP Peer Templates, page 33
- Inheritance in Peer Templates, page 33
- Peer Session Templates, page 34
- Peer Policy Templates, page 35
- BGP IPv6 Neighbor Activation Under the IPv4 Address Family, page 36

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS XE software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), and Virtual Private Networks version 4 (VPNv4).

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. Although BGP is referred to as an exterior gateway protocol (EGP) many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

Note

BGP requires more configuration than other routing protocols, and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.
BGP Router ID

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. By default, the Cisco IOS software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, then the software chooses the highest IPv4 address configured to a physical interface on the router to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

BGP-Speaker and Peer Relationships

A BGP-speaking router does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking routers. A peer device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a neighbor but, as this can imply the idea that the BGP devices are directly connected with no other router in between, the term neighbor will be avoided whenever possible in this document. A BGP speaker is the local router and a peer is any other BGP-speaking network device.

When a TCP connection is established between peers, each BGP peer initially exchanges all its routes—the complete BGP routing table—with the other peer. After this initial exchange only incremental updates are sent when there has been a topology change in the network, or when a routing policy has been implemented or modified. In the periods of inactivity between these updates, peers exchange special messages called keepalives.

A BGP autonomous system is a network controlled by a single technical administration entity. Peer routers are called external peers when they are in different autonomous systems and internal peers when they are in the same autonomous system. Usually, external peers are adjacent and share a subnet; internal peers may be anywhere in the same autonomous system.

For more details about external BGP peers, see the "Connecting to a Service Provider Using External BGP" module. For more details about internal BGP peers, see the "Configuring Internal BGP Features" module.

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were 2-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority (IANA) will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- **Asplain**—Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- **Asdot**—Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.
Asdot Only Autonomous System Number Formatting

In Cisco IOS XE Release 2.3, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers the asdot format includes a period, which is a special character in regular expressions. A backslash must be entered before the period (for example, \,14) to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in show command output in Cisco IOS images where only asdot formatting is available.

Table 6  Asdot Only 4-Byte Autonomous System Number Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

Asplain as Default Autonomous System Number Formatting

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you want to change the default show command output to display 4-byte autonomous system numbers in the asdot format, use the bgp asnotation dot command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte autonomous system numbers in either asplain or asdot format, only one format is used to display show command output and control 4-byte autonomous system number matching for regular expressions, and the default is asplain format. To display 4-byte autonomous system numbers in show command output and to control matching for regular expressions in the asdot format, you must configure the bgp asnotation dot command. After enabling the bgp asnotation dot command, a hard reset must be initiated for all BGP sessions by entering the clear ip bgp * command.

Note

If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The show command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

Table 7  Default Asplain 4-Byte Autonomous System Number Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asplain</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
</tr>
</tbody>
</table>
Reserved and Private Autonomous System Numbers

In Cisco IOS XE Release 2.3 and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.

Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous-system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: [http://www.iana.org/](http://www.iana.org/).

Cisco Implementation of 4-Byte Autonomous System Numbers

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain--65538 for example--as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression
match and output display of 4-byte autonomous system numbers to asdot format, use the `bgp asnotation dot` command followed by the `clear ip bgp *` command to perform a hard reset of all current BGP sessions. For more details about 4-byte autonomous system number formats, see the BGP Autonomous System Number Formats, page 23.

In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte autonomous system numbers uses asdot--1.2 for example--as the only configuration format, regular expression match, and output display, with no asplain support. For an example of BGP peers in two autonomous systems using 4-byte numbers, see the figure below. To view a configuration example of the configuration between three neighbor peers in separate 4-byte autonomous systems configured using asdot notation, see the Example Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers, page 103.

Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number be upgraded to support 4-byte autonomous system numbers.

---

A new private autonomous system number, 23456, was created by RFC 4893, and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

---

**Figure 5**  BGP Peers in Two Autonomous Systems Using 4-Byte Numbers

**BGP Peer Session Establishment**

When a BGP routing process establishes a peering session with a peer it goes through the following state changes:

- **Idle**--Initial state the BGP routing process enters when the routing process is enabled or when the router is reset. In this state, the router waits for a start event, such as a peering configuration with a remote peer. After the router receives a TCP connection request from a remote peer, the router initiates another start event to wait for a timer before starting a TCP connection to a remote peer. If the router is reset then the peer is reset and the BGP routing process returns to the Idle state.
• Connect--The BGP routing process detects that a peer is trying to establish a TCP session with the local BGP speaker.
• Active--In this state, the BGP routing process tries to establish a TCP session with a peer router using the ConnectRetry timer. Start events are ignored while the BGP routing process is in the Active state. If the BGP routing process is reconfigured or if an error occurs, the BGP routing process will release system resources and return to an Idle state.
• OpenSent--The TCP connection is established and the BGP routing process sends an OPEN message to the remote peer, and transitions to the OpenSent state. The BGP routing process can receive other OPEN messages in this state. If the connection fails, the BGP routing process transitions to the Active state.
• OpenReceive--The BGP routing process receives the OPEN message from the remote peer and waits for an initial keepalive message from the remote peer. When a keepalive message is received, the BGP routing process transitions to the Established state. If a notification message is received, the BGP routing process transitions to the Idle state. If an error or configuration change occurs that affects the peering session, the BGP routing process sends a notification message with the Finite State Machine (FSM) error code and then transitions to the Idle state.
• Established--The initial keepalive is received from the remote peer. Peering is now established with the remote neighbor and the BGP routing process starts exchanging update message with the remote peer. The hold timer restarts when an update or keepalive message is received. If the BGP process receives an error notification, it will transition to the Idle state.

Cisco Implementation of BGP Global and Address Family Configuration Commands

The address family model for configuring BGP is based on splitting apart the configuration for each address family. All commands that are independent of the address family are grouped together at the beginning (highest level) of the configuration, and these are followed by separate submodes for commands specific to each address family (with the exception that commands relating to IPv4 unicast can also be entered at the beginning of the configuration). When a network operator configures BGP, the flow of BGP configuration categories is represented by the following bullets in order:

• Global configuration--Configuration that is applied to BGP in general, rather than to specific neighbors. For example, the network, redistribute, and bgp bestpath commands.
• Address family-dependent configuration--Configuration that applies to a specific address family such as policy on an individual neighbor.

The relationship between BGP global and BGP address family-dependent configuration categories is shown in the table below.

<table>
<thead>
<tr>
<th>Table 9</th>
<th>Relationships Between BGP Configuration Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Configuration Category</td>
<td>Configuration Sets Within Category</td>
</tr>
<tr>
<td>Global address family-independent</td>
<td>One set of global address family-independent configurations</td>
</tr>
<tr>
<td>Address family-dependent</td>
<td>One set of global address family-dependent configurations per address family</td>
</tr>
</tbody>
</table>
Note

Address family configuration must be entered within the address family submode to which it applies.

The following is an example of BGP configuration statements showing the grouping of global address family-independent and address family-dependent commands.

```
router bgp <AS>
! AF independent part
neighbor <ip-address> <command> ! Session config; AF independent
   address-family ipv4 unicast
! AF dependant part
neighbor <ip-address> <command> ! Policy config; AF dependant
exit-address-family
address-family ipv4 multicast
! AF dependant part
neighbor <ip-address> <command> ! Policy config; AF dependant
exit-address-family
! VRF specific AS independent commands
! VRF specific AS dependant commands
neighbor <ip-address> <command> ! Session config; AF independent
neighbor <ip-address> <command> ! Policy config; AF dependant
exit-address-family
```

The following example shows actual BGP commands that match the BGP configuration statements in the previous example:

```
router bgp 45000
   router-id 172.17.1.99
   bgp log-neighbor-changes
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
   address-family ipv4 unicast
   neighbor 192.168.1.2 activate
   network 172.17.1.0 mask 255.255.255.0
exit-address-family
   address-family ipv4 multicast
   neighbor 192.168.3.2 activate
   network 172.16.1.0 mask 255.255.255.0
   exit-address-family
   address-family ipv4 unicast vrf <vrf-name>
   neighbor 192.168.3.2 activate
   network 172.21.1.0 mask 255.255.255.0
exit-address-family
```

In Cisco IOS XE Release 2.1 and later releases, the `bgp upgrade-cli` command simplifies the migration of BGP networks and existing configurations from the network layer reachability information (NLRI) format to the address family format. Network operators can configure commands in the address family identifier (AFI) format and save these command configurations to existing NLRI formatted configurations. The BGP hybrid command-line interface (CLI) does not add support for complete AFI and NLRI integration because of the limitations of the NLRI format. For complete support of AFI commands and features, we recommend upgrading existing NLRI configurations with the `bgp upgrade-cli` command. For a configuration example of migrating BGP configurations from the NLRI format to the address family format, see the GUID-F00AB703-23D5-4217-B406-1C3A8C5F64F0.

**BGP Session Reset**

Whenever there is a change in the routing policy due to a configuration change, BGP peering sessions must be reset using the `clear ip bgp` command. Cisco IOS XE software support the following three mechanisms to reset BGP peering sessions:
• Hard reset--A hard reset tears down the specified peering sessions including the TCP connection and deletes routes coming from the specified peer.
• Soft reset--A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reconfiguration uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply new BGP policy without disrupting the network. Soft reconfiguration can be configured for inbound or outbound sessions.
• Dynamic inbound soft reset--The route refresh capability, as defined in RFC 2918, allows the local router to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for non-disruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP routers must support the route refresh capability. To determine if a BGP router supports this capability, use the `show ip bgp neighbors` command. The following message is displayed in the output when the router supports the route refresh capability:

```
Received route refresh capability from peer.
```

The `bgp soft-reconfig-backup` command was introduced to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.

### BGP Route Aggregation

BGP peers store and exchange routing information and the amount of routing information increases as more BGP speakers are configured. The use of route aggregation reduces the amount of information involved. Aggregation is the process of combining the attributes of several different routes so that only a single route is advertised. Aggregate prefixes use the classless interdomain routing (CIDR) principle to combine contiguous networks into one classless set of IP addresses that can be summarized in routing tables. Fewer routes now need to be advertised.

Two methods are available in BGP to implement route aggregation. You can redistribute an aggregated route into BGP or you can use a form of conditional aggregation. Basic route redistribution involves creating an aggregate route and then redistributing the routes into BGP. Conditional aggregation involves creating an aggregate route and then advertising or suppressing the advertising of certain routes on the basis of route maps, autonomous system set path (AS-SET) information, or summary information.

The `bgp suppress-inactive` command configures BGP to not advertise inactive routes to any BGP peer. A BGP routing process can advertise routes that are not installed in the routing information database (RIB) to BGP peers by default. A route that is not installed into the RIB is an inactive route. Inactive route advertisement can occur, for example, when routes are advertised through common route aggregation. Inactive route advertisements can be suppressed to provide more consistent data forwarding.

### BGP Route Aggregation AS-SET Information Generation

AS-SET information can be generated when BGP routes are aggregated using the `aggregate-address` command. The path advertised for such a route is an AS-SET consisting of all the elements, including the communities, contained in all the paths that are being summarized. If the AS-PATHs to be aggregated are identical, only the AS-PATH is advertised. The ATOMIC-AGGREGATE attribute, set by default for the `aggregate-address` command, is not added to the AS-SET.
Routing Policy Change Management

Routing policies for a peer include all the configurations for elements such as route map, distribute list, prefix list, and filter list that may impact inbound or outbound routing table updates. Whenever there is a change in the routing policy, the BGP session must be soft cleared, or soft reset, for the new policy to take effect. Performing inbound reset enables the new inbound policy configured on the router to take effect. Performing outbound reset causes the new local outbound policy configured on the router to take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reset, a new inbound policy of the neighbor can also take effect. This means that after changing inbound policy you must do an inbound reset on the local router or an outbound reset on the peer router. Outbound policy changes require an outbound reset on the local router or an inbound reset on the peer router.

There are two types of reset: hard reset and soft reset. The table below lists their advantages and disadvantages.

<table>
<thead>
<tr>
<th>Table 10 Advantages and Disadvantages of Hard and Soft Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Reset</strong></td>
</tr>
<tr>
<td>Hard reset</td>
</tr>
<tr>
<td>Outbound soft reset</td>
</tr>
<tr>
<td>Dynamic inbound soft reset</td>
</tr>
<tr>
<td>Configured inbound soft reset (uses the <strong>neighbor soft-reconfiguration</strong> router configuration command)</td>
</tr>
</tbody>
</table>

Once you have defined two routers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you must reset BGP connections for the configuration change to take effect. A soft reset updates the routing table for inbound and outbound routing updates. Cisco IOS XE software supports soft reset without any prior configuration. This soft reset allows the dynamic exchange of route
refresh requests and routing information between BGP routers, and the subsequent readvertisement of the respective outbound routing table. There are two types of soft reset:

- When soft reset is used to generate inbound updates from a neighbor, it is called dynamic inbound soft reset.
- When soft reset is used to send a new set of updates to a neighbor, it is called outbound soft reset.

To use soft reset without preconfiguration, both BGP peers must support the soft route refresh capability, which is advertised in the OPEN message sent when the peers establish a TCP session.

### Conditional BGP Route Injection

Routes that are advertised through the BGP are commonly aggregated to minimize the number of routes that are used and reduce the size of global routing tables. However, common route aggregation can obscure more specific routing information that is more accurate but not necessary to forward packets to their destinations. Routing accuracy is obscured by common route aggregation because a prefix that represents multiple addresses or hosts over a large topological area cannot be accurately reflected in a single route. Cisco IOS XE software provides several methods in which you can originate a prefix into BGP. The existing methods include redistribution and using the `network` or `aggregate-address` command. These methods assume the existence of more specific routing information (matching the route to be originated) in either the routing table or the BGP table.

BGP conditional route injection allows you to originate a prefix into a BGP routing table without the corresponding match. This feature allows more specific routes to be generated based on administrative policy or traffic engineering information in order to provide more specific control over the forwarding of packets to these more specific routes, which are injected into the BGP routing table only if the configured conditions are met. Enabling this feature will allow you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix may be injected. BGP conditional route injection is enabled with the `bgp inject-map exist-map` command and uses two route maps (inject map and exist map) to install one (or more) more specific prefixes into a BGP routing table. The exist map specifies the prefixes that the BGP speaker will track. The inject map defines the prefixes that will be created and installed into the local BGP table.

### BGP Peer Groups

Often, in a BGP network, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into BGP peer groups to simplify configuration and, more importantly, to make configuration updates more efficient. When you have many peers, this approach is highly recommended.

### BGP Backdoor Routes

In a BGP network topology with two border routers using eBGP to communicate to a number of different autonomous systems, using eBGP to communicate between the two border routers may not be the most efficient routing method. In the figure below, Router B as a BGP speaker will receive a route to Router D through eBGP, but this route will traverse at least two autonomous systems. Router B and Router D are also connected through an Enhanced Interior Gateway Routing Protocol (EIGRP) network (any IGP can be used here) and this route has a shorter path. EIGRP routes, however, have a default administrative distance of 90 and eBGP routes have a default administrative distance of 20, so BGP will prefer the eBGP route. Changing the default administrative distances is not recommended because changing the administrative distance may lead to routing loops. To cause BGP to prefer the EIGRP route, you can use the `network backdoor` command. BGP treats the network specified by the `network backdoor` command as a locally
assigned network, except that it does not advertise the specified network in BGP updates. In the figure below, this means that Router B will communicate to Router D using the shorter EIGRP route instead of the longer eBGP route.

**Figure 6**  
_BGP Backdoor Route Topology_

---

**BGP Update Group**

The introduction of the BGP (dynamic) update group in Cisco IOS XE Release 2.1 provides a different type of BGP peer grouping from existing BGP peer groups. Existing peer groups are not affected but peers with the same outbound policy configured that are not members of a current peer group can be grouped into an update group. The members of this update group will use the same update generation engine. When BGP update groups are configured an algorithm dynamically calculates the BGP update group membership based on outbound policies. Optimal BGP update message generation occurs automatically and independently. BGP neighbor configuration is no longer restricted by outbound routing policies, and update groups can belong to different address families.

**BGP Dynamic Update Group Configuration**

In Cisco IOS XE Release 2.1 and later releases, an algorithm was introduced that dynamically calculates and optimizes update groups of neighbors that share the same outbound policies and can share the same update messages. No configuration is required to enable the BGP dynamic update group and the algorithm runs automatically. When a change to outbound policy occurs, the router automatically recalculates update group memberships and applies the changes by triggering an outbound soft reset after a 1-minute timer expires. This behavior is designed to provide the network operator with time to change the configuration if a mistake is made. You can manually enable an outbound soft reset before the timer expires by entering the `clear ip bgp ip-address soft out` command.

For the best optimization of BGP update group generation, we recommend that the network operator keeps outbound routing policy the same for neighbors that have similar outbound policies.
BGP Peer Templates

To address some of the limitations of peer groups such as configuration management, BGP peer templates were introduced to support the BGP update group configuration.

A peer template is a configuration pattern that can be applied to neighbors that share policies. Peer templates are reusable and support inheritance, which allows the network operator to group and apply distinct neighbor configurations for BGP neighbors that share policies. Peer templates also allow the network operator to define very complex configuration patterns through the capability of a peer template to inherit a configuration from another peer template.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and NLRI configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. BGP peer routers using peer templates also benefit from automatic update group configuration. With the configuration of the BGP peer templates and the support of the BGP dynamic update peer groups, the network operator no longer needs to configure peer groups in BGP and the network can benefit from improved configuration flexibility and faster convergence.

**Note**

A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

Inheritance in Peer Templates

The inheritance capability is a key component of peer template operation. Inheritance in a peer template is similar to node and tree structures commonly found in general computing, for example, file and directory trees. A peer template can directly or indirectly inherit the configuration from another peer template. The directly inherited peer template represents the tree in the structure. The indirectly inherited peer template represents a node in the tree. Because each node also supports inheritance, branches can be created that apply the configurations of all indirectly inherited peer templates within a chain back to the directly inherited peer template or the source of the tree. This structure eliminates the need to repeat configuration statements that are commonly reapplied to groups of neighbors because common configuration statements can be applied once and then indirectly inherited by peer templates that are applied to neighbor groups with common configurations. Configuration statements that are duplicated separately within a node and a tree are filtered out at the source of the tree by the directly inherited template. A directly inherited template will overwrite any indirectly inherited statements that are duplicated in the directly inherited template.

Inheritance expands the scalability and flexibility of neighbor configuration by allowing you to chain together peer templates configurations to create simple configurations that inherit common configuration statements or complex configurations that apply very specific configuration statements along with common
inherited configurations. Specific details about configuring inheritance in peer session templates and peer policy templates are provided in the following sections.

When BGP neighbors use inherited peer templates it can be difficult to determine which policies are associated with a specific template. In Cisco IOS XE Release 2.1 and later releases, the detail keyword was added to the `show ip bgp template peer-policy` command to display the detailed configuration of local and inherited policies associated with a specific template.

Peer Session Templates

Peer session templates are used to group and apply the configuration of general session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode.

Only general session commands can be configured in a peer session template. The following general session commands are supported by peer session templates:

- `description`
- `disable-connected-check`
- `ebgp-multihop`
- `exit peer-session`
- `inherit peer-session`
- `local-as`
- `password`
- `remote-as`
- `shutdown`
- `timers`
- `translate-update`
- `update-source`
- `version`

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template.

If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

This behavior allows a BGP neighbor to directly inherit only one session template and indirectly inherit up to seven additional peer session templates. This allows you to apply up to a maximum of eight peer session configurations to a neighbor: the configuration from the directly inherited peer session template and the configurations from up to seven indirectly inherited peer session templates. Inherited peer session configurations are evaluated first and applied starting with the last node in the branch and ending with the directly applied peer session template configuration at the source of the tree. The directly applied peer session template will have priority over inherited peer session template configurations. Any configuration statements that are duplicated in inherited peer session templates will be overwritten by the directly applied
peer session template. So, if a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template. The following examples illustrate the use of this feature.

In the following example, the general session command `remote-as 1` is applied in the peer session template named `SESSION-TEMPLATE-ONE`:

```
template peer-session SESSION-TEMPLATE-ONE
  remote-as 1
  exit peer-session
```

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

**Peer Policy Templates**

Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration mode. Peer policy templates are created and configured in peer policy configuration mode. BGP policy commands that are configured for specific address families are configured in a peer policy template. The following BGP policy commands are supported by peer policy templates:

- `advertisement-interval`
- `allowas-in`
- `as-overide`
- `capability`
- `default-originate`
- `distribute-list`
- `dmzlink-bw`
- `exit-peer-policy`
- `filter-list`
- `inherit peer-policy`
- `maximum-prefix`
- `next-hop-self`
- `next-hop-unchanged`
- `prefix-list`
- `remove-private-as`
- `route-map`
- `route-reflector-client`
- `send-community`
- `send-label`
- `soft-reconfiguration`
- `unsuppress-map`
- `weight`

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address families. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.
Like peer session templates, a peer policy template supports inheritance. However, there are minor differences. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group. Inherited peer policy templates are configured with sequence numbers like route maps. An inherited peer policy template, like a route map, is evaluated starting with the inherit statement with the lowest sequence number and ending with the highest sequence number. However, there is a difference; a peer policy template will not collapse like a route map. Every sequence is evaluated, and if a BGP policy command is reapplied with a different value, it will overwrite any previous value from a lower sequence number.

The directly applied peer policy template and the inherit statement with the highest sequence number will always have priority and be applied last. Commands that are reapplied in subsequent peer templates will always overwrite the previous values. This behavior is designed to allow you to apply common policy configurations to large neighbor groups and specific policy configurations only to certain neighbors and neighbor groups without duplicating individual policy configuration commands.

Peer policy templates support only policy configuration commands. BGP policy configuration commands that are configured only for specific address families are configured with peer policy templates.

The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, very specific and very complex BGP policies can also be created.

**BGP IPv6 Neighbor Activation Under the IPv4 Address Family**

Prior to Cisco IOS XE Release 3.3.1S, by default, both IPv6 and IPv4 capability is exchanged with a BGP peer that has an IPv6 address. When an IPv6 peer is configured, that neighbor is automatically activated under the IPv4 unicast address family.

Beginning with Cisco IOS XE Release 3.3.1S, when a new IPv6 neighbor is being configured, it is no longer automatically activated under the IPv4 address family. You can manually activate the IPv6 neighbor under the IPv4 address family if, for example, you have a dual stack environment and want to send IPv6 and IPv4 prefixes.

If you do not want an existing IPv6 peer to be activated under the IPv4 address family, you can manually deactivate the peer with the `no neighbor activate` command. Until then, existing configurations that activate an IPv6 neighbor under the IPv4 unicast address family will continue to try to establish a session.

**How to Configure a Basic BGP Network**

Configuring a basic BGP network consists of a few required tasks and many optional tasks. A BGP routing process must be configured and BGP peers must be configured, preferably using the address family configuration model. If the BGP peers are part of a VPN network, the BGP peers must be configured using the IPv4 VRF address family task. The other tasks in the following list are optional:

- Configuring a BGP Routing Process, page 37
- Configuring a BGP Peer, page 40
- Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers, page 43
- Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers, page 46
- Configuring a BGP Peer for the IPv4 VRF Address Family, page 50
- Customizing a BGP Peer, page 54
Configuring a BGP Routing Process

Perform this task to configure a BGP routing process. You must perform the required steps at least once to enable BGP. The optional steps here allow you to configure additional features in your BGP network. Several of the features, such as logging neighbor resets and immediate reset of a peer when its link goes down, are enabled by default but are presented here to enhance your understanding of how your BGP network operates.

Note

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple concurrent BGP address family and subaddress family configurations.

The configuration in this task is done at Router A in the figure below and would need to be repeated with appropriate changes to the IP addresses (for example, at Router B) to fully achieve a BGP process between the two routers. No address family is configured here for the BGP routing process so routing information for the IPv4 unicast address family is advertised by default.

Figure 7  BGP Topology with Two Autonomous Systems
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. network network-number [mask network-mask][route-map route-map-name]
5. bgp router-id ip-address
6. timers bgp keepalive holdtime
7. bgp fast-external-fallover
8. bgp log-neighbor-changes
9. end
10. show ip bgp [network] [network-mask]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Configures a BGP routing process, and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 40000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> network network-number [mask network-mask][route-map route-map-name]</td>
<td>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# network 10.1.1.0 mask 255.255.255.0</td>
<td></td>
</tr>
</tbody>
</table>

For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
### Command or Action

<table>
<thead>
<tr>
<th>Step 5 bgp router-id <em>ip-address</em></th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:** Router(config-router)# bgp router-id 10.1.1.99 | (Optional) Configures a fixed 32-bit router ID as the identifier of the local router running BGP.  
  - Use the *ip-address* argument to specify a unique router ID within the network.  
  
  **Note** Configuring a router ID using the `bgp router-id` command resets all active BGP peering sessions. |

<table>
<thead>
<tr>
<th>Step 6 timers bgp <em>keepalive</em> holdtime</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:** Router(config-router)# timers bgp 70 120 | (Optional) Sets BGP network timers.  
  - Use the *keepalive* argument to specify the frequency, in seconds, with which the software sends keepalive messages to its BGP peer. By default, the keepalive timer is set to 60 seconds.  
  - Use the *holdtime* argument to specify the interval, in seconds, after not receiving a keepalive message that the software declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds. |

<table>
<thead>
<tr>
<th>Step 7 bgp fast-external-fallover</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:** Router(config-router)# bgp fast-external-fallover | (Optional) Enables the automatic resetting of BGP sessions.  
  - By default, the BGP sessions of any directly adjacent external peers are reset if the link used to reach them goes down. |

<table>
<thead>
<tr>
<th>Step 8 bgp log-neighbor-changes</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:** Router(config-router)# bgp log-neighbor-changes | (Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets.  
  - Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated. |

<table>
<thead>
<tr>
<th>Step 9 end</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Router(config-router)# end</td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10 <strong>show ip bgp [network] [network-mask]</strong></th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Example:** Router# show ip bgp | (Optional) Displays the entries in the BGP routing table.  
  **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*. |
Examples

The following sample output from the `show ip bgp` command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A. You can see an entry for the network 10.1.1.0 that is local to this autonomous system.

```
BGP table version is 12, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
```

Troubleshooting Tips

Use the `ping` command to check basic network connectivity between the BGP routers.

Configuring a BGP Peer

Perform this task to configure BGP between two IPv4 routers (peers). The address family configured here is the default IPv4 unicast address family and the configuration is done at Router A in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before you perform this task, perform the GUID-234727BA-098D-4492-AD2A-3FD8D4C6EFE9 task.

Note

By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types, such as IPv6 prefixes.

---

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. address-family ipv4 [unicast | multicast] vrf vrf-name
6. neighbor ip-address activate
7. end
8. show ip bgp [network] [network-mask]
9. show ip bgp neighbors [neighbor-address]
### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 40000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.1.1 remote-as 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family ipv4 [unicast</td>
<td>multicast] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address activate</td>
<td>Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-af)# neighbor 192.168.1.1 activate</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# end</td>
</tr>
<tr>
<td><strong>Step 8</strong> show ip bgp [network] [network-mask]</td>
<td>(Optional) Displays the entries in the BGP routing table. <strong>Note</strong> Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show ip bgp</td>
</tr>
<tr>
<td><strong>Step 9</strong> show ip bgp neighbors [neighbor-address]</td>
<td>(Optional) Displays information about the TCP and BGP connections to neighbors. <strong>Note</strong> Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# show ip bgp neighbors 192.168.2.2</td>
</tr>
</tbody>
</table>

**Examples**

The following sample output from the `show ip bgp` command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A and Router B. You can now see an entry for the network 172.17.1.0 in autonomous system 45000.

```
BGP table version is 13, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop  Metric LocPrf Weight Path
*> 10.1.1.0/24 0.0.0.0 0 32768 i
*> 172.17.1.0/24 192.168.1.1 0 0 45000 i
```

The following sample output from the `show ip bgp neighbors` command shows information about the TCP and BGP connections to the BGP neighbor 192.168.1.1 of Router A in the figure above after this task has been configured on Router A:

```
BGP neighbor is 192.168.1.1, remote AS 45000, external link
BGP version 4, remote router ID 172.17.1.99
BGP state - Established, up for 00:06:55
Last read 00:00:15, last write 00:00:15, hold time is 120, keepalive intervals
Configured hold time is 120, keepalive interval is 70 seconds, Minimum holdtimes
Neighbor capabilities:
  Route refresh: advertised and received (old & new)
  Address family IPv4 Unicast: advertised and received
Message statistics:
  InQ depth is 0
  OutQ depth is 0
  Sent  Rcvd
  Opens:  1  1
  Notifications:  0  0
  Updates:  1  2
  Keepalives:  13 13
  Route Refresh:  0  0
  Total:  15  16
Default minimum time between advertisement runs is 30 seconds
```
For address family: IPv4 Unicast
BGP table version 13, neighbor version 13/0
Output queue size : 0
Index 1, Offset 0, Mask 0x2
1 update-group member

Prefix activity: Sent  Rcvd
Prefixes Current: 1 1 (Consumes 52 bytes)
Prefixes Total: 1 1
Implicit Withdraw: 0 0
Explicit Withdraw: 0 0
Used as bestpath: n/a 1
Used as multipath: n/a 0

Local Policy Denied Prefixes: Outbound  Inbound
AS_PATH loop: n/a 1
Bestpath from this peer: 1 n/a
Total: 1 1
Number of NLRIs in the update sent: max 0, min 0
Connections established 1; dropped 0

Troubleshooting Tips
Use the ping command to verify basic network connectivity between the BGP routers.

What to Do Next
If you have BGP peers in a VPN, proceed to the Configuring a BGP Peer for the IPv4 VRF Address Family, page 50. If you do not have BGP peers in a VPN, proceed to the GUID-D8DCE04C-0F64-43B8-8649-25497EB8DCFC.

Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers
Perform this task to configure a BGP routing process and BGP peers when the BGP peers are located in 4-byte autonomous system numbers. The address family configured here is the default IPv4 unicast address
family, and the configuration is done at Router B in the figure above. The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

This task requires Cisco IOS XE Release 2.4 or a later release to be running on the router.

---

**Note**

By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types.

---

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. Repeat Step 4 to define other BGP neighbors, as required.
6. address-family ipv4 [unicast | multicast] vrf vrf-name
7. neighbor ip-address activate
8. Repeat Step 7 to activate other BGP neighbors, as required.
9. network network-number [mask network-mask] [route-map route-map-name]
10. end
11. show ip bgp [network] [network-mask]
12. show ip bgp summary

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 65538</td>
<td><strong>•</strong> In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.</td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.1.2 remote-as 65536</td>
<td><strong>•</strong> In this example, the 4-byte autonomous system number, 65536, is defined in asplain notation.</td>
</tr>
<tr>
<td><strong>Step 5</strong> Repeat Step 4 to define other BGP neighbors, as required.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 6</strong> address-family ipv4 [unicast</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4 unicast</td>
<td><strong>•</strong> The <strong>unicast</strong> keyword specifies the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the <strong>unicast</strong> keyword is not specified with the <strong>address-family ipv4</strong> command.</td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor ip-address activate</td>
<td>Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-af)# neighbor 192.168.1.2 activate</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 8</strong> Repeat Step 7 to activate other BGP neighbors, as required.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 9</strong> network network-number [mask network-mask] [route-map route-map-name]</td>
<td>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# network 172.17.1.0 mask 255.255.255.0</td>
<td><strong>•</strong> For exterior protocols the <strong>network</strong> command controls which networks are advertised. Interior protocols use the <strong>network</strong> command to determine where to send updates.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 10</th>
<th><strong>end</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# end
```

<table>
<thead>
<tr>
<th>Step 11</th>
<th><strong>show ip bgp [network] [network-mask]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>(Optional) Displays the entries in the BGP routing table.</td>
</tr>
</tbody>
</table>

**Note**  Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*. |

**Example:**

```
Router# show ip bgp 10.1.1.0
```

<table>
<thead>
<tr>
<th>Step 12</th>
<th><strong>show ip bgp summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>(Optional) Displays the status of all BGP connections.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router# show ip bgp summary
```

### Examples

The following output from the `show ip bgp` command at Router B shows the BGP routing table entry for network 10.1.1.0 learned from the BGP neighbor at 192.168.1.2 in Router A in the figure above with its 4-byte autonomous system number of 65536 displayed in the default asplain format.

```
RouterB# show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
   Advertised to update-groups:
   2
   65536
   192.168.1.12 from 192.168.1.12 (10.1.1.99)
   Origin IGP, metric 0, localpref 100, valid, external, best
```

The following output from the `show ip bgp summary` command shows the 4-byte autonomous system number 65536 for the BGP neighbor 192.168.1.2 of Router A in the figure above after this task has been configured on Router B:

- **Troubleshooting Tips, page 43**

### Troubleshooting Tips

Use the `ping` command to verify basic network connectivity between the BGP routers.

### Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers

Perform this task to modify the default output format for 4-byte autonomous system numbers from asplain format to asdot notation format. The `show ip bgp summary` command is used to display the changes in output format for the 4-byte autonomous system numbers.
For more details about 4-byte autonomous system number formats, see the BGP Autonomous System Number Formats, page 23.

This example requires Cisco IOS XE Release 2.4 or a later release, to be running on the router.

**SUMMARY STEPS**

1. `enable`
2. `show ip bgp summary`
3. `configure terminal`
4. `router bgp autonomous-system-number`
5. `bgp asnotation dot`
6. `end`
7. `clear ip bgp *`
8. `show ip bgp summary`
9. `show ip bgp regexp regexp`
10. `configure terminal`
11. `router bgp autonomous-system-number`
12. `no bgp asnotation dot`
13. `end`
14. `clear ip bgp *`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> show ip bgp summary</td>
<td>Displays the status of all BGP connections.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp summary</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 4** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process.  
  - In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.  
  **Example:**  
  Router(config)# router bgp 65538 |
| **Step 5** bgp asnotation dot | Changes the default output format of BGP 4-byte autonomous system numbers from asplain (decimal values) to dot notation.  
  **Note** 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for `show` commands or the matching of regular expressions.  
  **Example:**  
  Router(config-router)# bgp asnotation dot |
| **Step 6** end | Exits address family configuration mode and returns to privileged EXEC mode.  
  **Example:**  
  Router(config-router)# end |
| **Step 7** clear ip bgp * | Clears and resets all current BGP sessions.  
  - In this example, a hard reset is performed to ensure that the 4-byte autonomous system number format change is reflected in all BGP sessions.  
  **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.  
  **Example:**  
  Router# clear ip bgp * |
| **Step 8** show ip bgp summary | Displays the status of all BGP connections.  
  **Example:**  
  Router# show ip bgp summary |
| **Step 9** show ip bgp regexp regexp | Displays routes that match the autonomous system path regular expression.  
  - In this example, a regular expression to match a 4-byte autonomous system path is configured using asdot format.  
  **Example:**  
  Router# show ip bgp regexp ^1\\..0$ |
| **Step 10** configure terminal | Enters global configuration mode.  
  **Example:**  
  Router# configure terminal |
### Command or Action

**Step 11**  router bgp autonomous-system-number

**Example:**

```
Router(config)# router bgp 65538
```

Enters router configuration mode for the specified routing process.

- In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.

**Step 12**  no bgp asnotation dot

**Example:**

```
Router(config-router)# no bgp asnotation dot
```

Resets the default output format of BGP 4-byte autonomous system numbers back to asplain (decimal values).

**Note** 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for `show` commands or the matching of regular expressions.

**Step 13**  end

**Example:**

```
Router(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

**Step 14**  clear ip bgp *

**Example:**

```
Router# clear ip bgp *
```

Clears and resets all current BGP sessions.

- In this example, a hard reset is performed to ensure that the 4-byte autonomous system number format change is reflected in all BGP sessions.

**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IPRouting: BGP Command Reference*.

### Examples

The following output from the `show ip bgp summary` command shows the default asplain format of the 4-byte autonomous system numbers. Note the asplain format of the 4-byte autonomous system numbers, 65536 and 65550.

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor     V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2  4   65536   7   7   1   0 00:03:04      0
192.168.3.2  4   65550   4   4   1   0 00:00:15      0
```

After the `bgp asnotation dot` command is configured (followed by the `clear ip bgp *` command to perform a hard reset of all current BGP sessions), the output is converted to asdot notation format as shown in the following output from the `show ip bgp summary` command. Note the asdot format of the 4-byte autonomous system numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 autonomous system numbers).

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor     V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
```

**IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2**
After the `bgp asnotation dot` command is configured (followed by the `clear ip bgp *` command to perform a hard reset of all current BGP sessions), the regular expression match format for 4-byte autonomous system paths is changed to asdot notation format. Although a 4-byte autonomous system number can be configured in a regular expression using either asplain format or asdot format, only 4-byte autonomous system numbers configured using the current default format are matched. In the first example below, the `show ip bgp regexp` command is configured with a 4-byte autonomous system number in asplain format.

The match fails because the default format is currently asdot format and there is no output. In the second example using asdot format, the match passes and the information about the 4-byte autonomous system path is shown using the asdot notation.

---

**Note**

The asdot notation uses a period which is a special character in Cisco regular expressions. To remove the special meaning, use a backslash before the period.

---

**Configuring a BGP Peer for the IPv4 VRF Address Family**

Perform this optional task to configure BGP between two IPv4 routers (peers) that must exchange IPv4 VRF information because they exist in a VPN. The address family configured here is the IPv4 VRF address family and the configuration is done at Router B in the figure below with the neighbor 192.168.3.2 at Router E in autonomous system 50000. Remember to perform this task for any neighbor routers that are to be BGP IPv4 VRF address family peers.

This task does not show the complete configuration required for VPN routing. For some complete example configurations and an example configuration showing how to create a VRF with a route-target that uses a 4-byte autonomous system number, see the Example Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number, page 105.

**Figure 8**  
BGP Topology for IPv4 VRF Address Family
Before you perform this task, perform the GUID-234727BA-098D-4492-AD2A-3FD8D4C6EFE9 task.

**SUMMARY STEPS**

1. enable  
2. configure terminal  
3. `interface type number`  
4. `vrf forwarding vrf-name`  
5. `ip address ip-address mask [secondary [vrf vrf-name]]`  
6. exit  
7. `ip vrf vrf-name`  
8. `rd route-distinguisher`  
9. `route-target {import | export| both} route-target-ext-community`  
10. exit  
11. `router bgp autonomous-system-number`  
12. `address-family ipv4 [unicast | multicast| vrf vrf-name]`  
13. `neighbor ip-address remote-as autonomous-system-number`  
14. `neighbor {ip-address| peer-group-name} maximum-prefix maximum [threshold] [restart restart-interval] [warning-only]`  
15. `neighbor ip-address activate`  
16. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
   • Enter your password if prompted. |
| **Example:** |  
   Router> enable |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:** |  
   Router# configure terminal |
| **Step 3** `interface type number` | Enters interface configuration mode. |
| **Example:** |  
   Router(config)# interface GigabitEthernet 1/0/0 |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> vrf forwarding vrf-name</td>
<td>Associates a VPN VRF instance with an interface or subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# vrf forwarding vpn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ip address ip-address mask [secondary [vrf vrf-name]]</td>
<td>Sets an IP address for an interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ip address 192.168.3.1 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> ip vrf vrf-name</td>
<td>Configures a VRF routing table and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Use the vrf-name argument to specify a name to be assigned to the VRF.</td>
</tr>
<tr>
<td>Router(config)# ip vrf vpn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> rd route-distinguisher</td>
<td>Creates routing and forwarding tables and specifies the default route distinguisher for a VPN.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Use the route-distinguisher argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.</td>
</tr>
<tr>
<td>Router(config-vrf)# rd 45000:5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Use the import keyword to import routing information from the target VPN extended community.</td>
</tr>
<tr>
<td>Router(config-vrf)# route-target both 45000:100</td>
<td>• Use the export keyword to export routing information to the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td>• Use the both keyword to import both import and export routing information to the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td>• Use the route-target-ext-community argument to add the route target extended community attributes to the VRF’s list of import, export, or both (import and export) route target extended communities.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits VRF configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)# exit</td>
</tr>
<tr>
<td><strong>Step 11</strong> ( \text{router bgp } \text{autonomous-system-number} )</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config)# router bgp 45000</td>
</tr>
<tr>
<td><strong>Step 12</strong> ( \text{address-family ipv4 [unicast</td>
<td>multicast]} \text{vrf vrf-name} )</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>unicast</strong> keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the <strong>unicast</strong> keyword is not specified with the <strong>address-family ipv4</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>multicast</strong> keyword to specify IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>vrf</strong> keyword and <strong>vrf-name</strong> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-router)# address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td><strong>Step 13</strong> ( \text{neighbor } \text{ip-address remote-as autonomous-system-number} )</td>
<td>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 192.168.3.2 remote-as 50000</td>
</tr>
<tr>
<td><strong>Step 14</strong> ( \text{neighbor [ip-address</td>
<td>peer-group-name]} \text{maximum-prefix maximum [threshold]} \text{[restart restart-interval]} [\text{warning-only}] )</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>maximum</strong> argument to specify the maximum number of prefixes allowed from the specified neighbor. The number of prefixes that can be configured is limited only by the available system resources on a router.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>threshold</strong> argument to specify an integer representing a percentage of the maximum prefix limit at which the router starts to generate a warning message.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>warning-only</strong> keyword to allow the router to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 192.168.3.2 maximum-prefix 10000 warning-only</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 15</th>
<th>neighbor ip-address activate</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# neighbor 192.168.3.2 activate
```

<table>
<thead>
<tr>
<th>Step 16</th>
<th>end</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exits address family configuration mode and enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# end
```

### Troubleshooting Tips

- Troubleshooting Tips, page 54

**Troubleshooting Tips**

Use the `ping vrf` command to verify basic network connectivity between the BGP routers, and use the `show ip vrf` command to verify that the VRF instance has been created.

**Customizing a BGP Peer**

Perform this task to customize your BGP peers. Although many of the steps in this task are optional, this task demonstrates how the neighbor and address family configuration command relationships work. Using the example of the IPv4 multicast address family, neighbor address family-independent commands are configured before the IPv4 multicast address family is configured. Commands that are address family-dependent are then configured and the `exit address-family` command is shown. An optional step shows how to disable a neighbor.
The configuration in this task is done at Router B in the figure below and would need to be repeated with appropriate changes to the IP addresses, for example, at Router E to fully configure a BGP process between the two routers.

**Figure 9  BGP Peer Topology**

By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types, such as IPv6 prefixes.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. no bgp default ipv4-unicast
5. neighbor (ip-address | peer-group-name) remote-as autonomous-system-number
6. neighbor (ip-address | peer-group-name) description text
7. address-family ipv4 [unicast | multicast | vrf vrf-name]
8. network network-number [mask network-mask] [route-map route-map-name]
9. neighbor (ip-address | peer-group-name) activate
10. neighbor (ip-address | peer-group-name) advertisement-interval seconds
11. neighbor (ip-address | peer-group-name) default-originate [route-map map-name]
12. exit-address-family
13. neighbor (ip-address | peer-group-name) shutdown
14. end
15. show ip bgp ipv4 multicast [command]
16. show ip bgp neighbors [neighbor-address] [received-routes | routes | advertised-routes | paths regexp | dampened-routes | received prefix-filter]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>Disables the IPv4 unicast address family for the BGP routing process. <strong>Note</strong> Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the <code>neighbor remote-as</code> router configuration command unless you configure the <code>no bgp default ipv4-unicast</code> router configuration command before configuring the <code>neighbor remote-as</code> command. Existing neighbor configurations are not affected.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor *(ip-address</td>
<td>peer-group-name)* remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.3.2 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor *(ip-address</td>
<td>peer-group-name)* description text</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.3.2 description finance</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> address-family ipv4 [unicast</td>
<td>multicast] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4 multicast</td>
<td>• The <em>unicast</em> keyword specifies the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the <em>unicast</em> keyword is not specified with the <code>address-family ipv4</code> command.</td>
</tr>
<tr>
<td><strong>Step 8</strong> network network-number [mask network-mask] [route-map route-map-name]</td>
<td><em>(Optional)</em> Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</td>
<td>• For exterior protocols the <em>network</em> command controls which networks are advertised. Interior protocols use the <em>network</em> command to determine where to send updates.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor {ip-address</td>
<td>peer-group-name} activate</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.3.2 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> neighbor {ip-address</td>
<td>peer-group-name} advertisement-interval seconds</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> neighbor {ip-address</td>
<td>peer-group-name} default-originate [route-map map-name]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.3.2 default-originate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> exit-address-family</td>
<td>Exits address family configuration mode and enters router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# exit-address-family</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> neighbor {ip-address</td>
<td>peer-group-name} shutdown</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If you perform this step you will not be able to run either of the subsequent show command steps because you have disabled the neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.3.2 shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> end</td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# end</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 15
show ip bgp ipv4 multicast [command]

**Example:**

```bash
Router# show ip bgp ipv4 multicast
```

(Optional) Displays IPv4 multicast database-related information.

- Use the `command` argument to specify any multiprotocol BGP command that is supported. To see the supported commands, use the `?` prompt on the CLI.

#### Step 16
show ip bgp neighbors [neighbor-address] [received-routes | routes | advertised-routes | paths regexp | dampened-routes | received prefix-filter]]

**Example:**

```bash
Router# show ip bgp neighbors 192.168.3.2
```

(Optional) Displays information about the TCP and BGP connections to neighbors.

### Examples

The following sample output from the **show ip bgp ipv4 multicast** command shows BGP IPv4 multicast information for Router B in the figure above after this task has been configured on Router B and Router E. Note that the networks local to each router that were configured under IPv4 multicast address family appear in the output table.

BGP table version is 3, local router ID is 172.17.1.99  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - incomplete  

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.2.2.0/24</td>
<td>192.168.3.2</td>
<td>0</td>
<td>0</td>
<td>50000</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 172.17.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following partial sample output from the **show ip bgp neighbors** command for neighbor 192.168.3.2 shows general BGP information and specific BGP IPv4 multicast address family information about the neighbor. The command was entered on Router B in the figure above after this task had been configured on Router B and Router E.

BGP neighbor is 192.168.3.2, remote AS 50000, external link  
Description: finance  
BGP version 4, remote router ID 10.2.2.99  
BGP state = Established, up for 01:48:27  
Last read 00:00:26, last write 00:00:26, hold time is 120, keepalive intervals  
Configured hold time is 120, keepalive interval is 70 seconds, Minimum holdtime  
Neighbor capabilities:  
Route refresh: advertised and received (old & new)  
Address family IPv4 Unicast: advertised  
Address family IPv4 Multicast: advertised and received  

| Prefix activity:  
| Prefixes Current: | 1 | (Consumes 48 bytes)  
| Prefixes Total: | 1 |  
| Implicit Withdraw: | 0 | 0 |
Removing BGP Configuration Commands Using a Redistribution

BGP CLI configuration can become quite complex even in smaller BGP networks. If you need to remove any CLI configuration, you must consider all the implications of removing the CLI. Analyze the current running configuration to determine the current BGP neighbor relationships, any address family considerations, and even other routing protocols that are configured. Many BGP CLI commands affect other parts of the CLI configuration.

Perform this task to remove all the BGP configuration commands used in a redistribution of BGP routes into EIGRP. A route map can be used to match and set parameters or to filter the redistributed routes to ensure that routing loops are not created when these routes are subsequently advertised by EIGRP. When removing BGP configuration commands you must remember to remove or disable all the related commands. In this example, if the `route-map` command is omitted, then the redistribution will still occur and possibly with unexpected results as the route map filtering has been removed. Omitting just the `redistribute` command would mean that the route map is not applied, but it would leave unused commands in the running configuration.

For more details on BGP CLI removal, see the "BGP CLI Removal Considerations" concept in the "Cisco BGP Overview" module.

To view the redistribution configuration before and after the CLI removal, see the Examples Removing BGP Configuration Commands Using a Redistribution Example, page 108.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. no route-map `map-name`
4. router eigrp `autonomous-system-number`
5. no redistribute `protocol [as-number]`
6. end
7. show running-config
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> no route-map map-name</td>
<td>Removes a route map from the running configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td>• In this example, a route map named bgp-to-eigrp is removed from the configuration.</td>
</tr>
<tr>
<td>Router(config)# no route-map bgp-to-eigrp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> router eigrp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router eigrp 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> no redistribute protocol [as-number]</td>
<td>Disables the redistribution of routes from one routing domain into another routing domain.</td>
</tr>
<tr>
<td>Example:</td>
<td>• In this example, the configuration of the redistribution of BGP routes into the EIGRP routing process is removed from the running configuration.</td>
</tr>
<tr>
<td>Router(config-router)# no redistribute bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If a route map was included in the original <code>redistribute</code> command configuration, remember to remove the <code>route-map</code> command configuration as in Step 3 in this example task.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only the syntax applicable to this task is used in this example. For more details, see the <em>Cisco IOS IP Routing: BGP Command Reference</em>.</td>
</tr>
<tr>
<td><strong>Step 6</strong> end</td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# end</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 7**  
show running-config

### Purpose

(Optional) Displays the current running configuration on the router.

- Use this command to verify that the redistribute and route-map commands are removed from the router configuration.

#### Example:

```text
Router# show running-config
```

---

## Monitoring and Maintaining Basic BGP

The tasks in this section are concerned with the resetting and display of information about basic BGP processes and peer relationships. Once you have defined two routers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you may have to reset BGP connections for the configuration change to take effect.

- Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing, page 62
- Resetting and Displaying Basic BGP Information, page 65

### Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing

Perform this task to configure inbound soft reconfiguration using the `bgp soft-reconfig-backup` command for BGP peers that do not support the route refresh capability. BGP Peers that support the route refresh capability are unaffected by the configuration of this command.

#### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. bgp log-neighbor-changes
5. bgp soft-reconfig-backup
6. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
7. neighbor {ip-address | peer-group-name} soft-reconfiguration{inbound}
8. neighbor {ip-address | peer-group-name} route-map map-name{in | out}
9. Repeat Steps 6 through 8 for every peer that is to be configured with soft-reconfiguration inbound.
10. exit
11. route-map map-name [permit|deny][sequence-number]
12. set local-preference number-value
13. end
14. show ip bgp neighbors [neighbor-address]
15. show ip bgp [network] [network-mask]
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| **Example:** Router(config)# router bgp 45000 |
| **Step 4** bgp log-neighbor-changes | Enables logging of BGP neighbor resets. |
| **Example:** Router(config-router)# bgp log-neighbor-changes |
| **Step 5** bgp soft-reconfig-backup | Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability.  
• This command is used to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command. |
<p>| <strong>Example:</strong> Router(config-router)# bgp soft-reconfig-backup |
| <strong>Step 6</strong> neighbor {ip-address | peer-group-name} remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. |
| <strong>Example:</strong> Router(config-router)# neighbor 192.168.1.2 remote-as 40000 |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 7** neighbor {ip-address | peer-group-name} soft-reconfiguration[inbound] | Configures the Cisco IOS XE software to start storing updates.  
• All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates. |
| *Example:*                                              |                                                                                                                                             |
| Router(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound |                                                                                                                                                     |
| **Step 8** neighbor {ip-address | peer-group-name} route-map map-name [in | out] | Applies a route map to incoming or outgoing routes.  
• In this example, the route map named LOCAL will be applied to incoming routes. |
| *Example:*                                              |                                                                                                                                             |
| Router(config-router)# neighbor 192.168.1.2 route-map LOCAL in |                                                                                                                                                     |
| **Step 9** Repeat Steps 6 through 8 for every peer that is to be configured with soft-reconfiguration inbound. | --                                                                                                                                         |
| **Step 10** exit                                        | Exits router configuration mode and enters global configuration mode.                                                                                   |
| *Example:*                                              |                                                                                                                                             |
| Router(config-router)# exit |                                                                                                                                                     |
| **Step 11** route-map map-name {permit | deny}[sequence-number] | Configures a route map and enters route map configuration mode.  
• In this example, a route map named LOCAL is created. |
| *Example:*                                              |                                                                                                                                             |
| Router(config)# route-map LOCAL permit 10               |                                                                                                                                                     |
| **Step 12** set local-preference number-value           | Specifies a preference value for the autonomous system path.  
• In this example, the local preference value is set to 200. |
| *Example:*                                              |                                                                                                                                             |
| Router(config-route-map)# set local-preference 200      |                                                                                                                                                     |
| **Step 13** end                                         | Exits route map configuration mode and enters privileged EXEC mode.  
•                                                                                                                                                     |
| *Example:*                                              |                                                                                                                                             |
| Router(config-route-map)# end                           |                                                                                                                                                     |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td><code>show ip bgp neighbors</code> [neighbor-address]</td>
<td>(Optional) Displays information about the TCP and BGP connections to neighbors.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-router-af)# show ip bgp neighbors 192.168.1.2</code></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><code>show ip bgp</code> [network] [network-mask]</td>
<td>(Optional) Displays the entries in the BGP routing table.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router# show ip bgp</code></td>
<td></td>
</tr>
</tbody>
</table>

### Examples

The following partial output from the `show ip bgp neighbors` command shows information about the TCP and BGP connections to the BGP neighbor 192.168.2.1. This peer supports route refresh.

```
BGP neighbor is 192.168.1.2, remote AS 40000, external link
Neighbor capabilities:
  Route refresh: advertised and received(new)
```

The following partial output from the `show ip bgp neighbors` command shows information about the TCP and BGP connections to the BGP neighbor 192.168.3.2. This peer does not support route refresh so the soft-reconfig inbound paths for BGP peer 192.168.3.2 will be stored because there is no other way to update any inbound policy updates.

```
BGP neighbor is 192.168.3.2, remote AS 50000, external link
Neighbor capabilities:
  Route refresh: advertised
```

The following sample output from the `show ip bgp` command shows the entry for the network 172.17.1.0. Both BGP peers are advertising 172.17.1.0/24 but only the received-only path is stored for 192.168.3.2.

```
BGP routing table entry for 172.17.1.0/24, version 11
Paths: (3 available, best #3, table Default-IP-Routing-Table, RIB-failure(4))
Flag: 0x820
  Advertised to update-groups:
    1
      50000
        192.168.3.2 from 192.168.3.2 (172.17.1.0)
          Origin incomplete, metric 0, localpref 200, valid, external
        50000, (received-only)
          192.168.3.2 from 192.168.3.2 (172.17.1.0)
            Origin incomplete, metric 0, localpref 100, valid, external
        40000
          192.168.1.2 from 192.168.1.2 (172.16.1.0)
            Origin incomplete, metric 6, localpref 200, valid, external, best
```

### Resetting and Displaying Basic BGP Information

Perform this task to reset and display information about basic BGP processes and peer relationships.
SUMMARY STEPS

1. enable
2. clear ip bgp { * | autonomous-system-number | neighbor-address } [ soft [ in | out ] ]
3. show ip bgp [ network-address | network-mask | longer-prefixes ] [ prefix-list prefix-list-name | route-map route-map-name ] [ shorter prefixes mask-length ]
4. show ip bgp neighbors [ neighbor-address ] [ received-routes | routes | advertised-routes | paths regexp | dampened-routes | received prefix-filter ]
5. show ip bgp paths
6. show ip bgp summary

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> clear ip bgp { *</td>
<td>autonomous-system-number</td>
</tr>
<tr>
<td>Example:</td>
<td>In the example provided, all BGP neighbor sessions are cleared and reset.</td>
</tr>
<tr>
<td>Router# clear ip bgp *</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> show ip bgp [ network-address</td>
<td>network-mask</td>
</tr>
<tr>
<td>Example:</td>
<td>In the example provided, the BGP routing table information for the 10.1.1.0 network is displayed.</td>
</tr>
<tr>
<td>Router# show ip bgp 10.1.1.0 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> show ip bgp neighbors [ neighbor-address ] [ received-routes</td>
<td>routes</td>
</tr>
<tr>
<td>Example:</td>
<td>In the example provided, the routes advertised from the router to BGP neighbor 192.168.3.2 on another router are displayed.</td>
</tr>
<tr>
<td>Router# show ip bgp neighbors 192.168.3.2 advertised-routes</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 5</strong> show ip bgp paths</td>
<td>Displays information about all the BGP paths in the database.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp paths</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show ip bgp summary</td>
<td>Displays information about the status of all BGP connections.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp summary</td>
<td></td>
</tr>
</tbody>
</table>

**Aggregating Route Prefixes Using BGP**

BGP peers exchange information about local networks but this can quickly lead to large BGP routing tables. CIDR enables the creation of aggregate routes (or supernets) to minimize the size of routing tables. Smaller BGP routing tables can reduce the convergence time of the network and improve network performance. Aggregated routes can be configured and advertised using BGP. Some aggregations advertise only summary routes and other methods of aggregating routes allow more specific routes to be forwarded. Aggregation applies only to routes that exist in the BGP routing table. An aggregated route is forwarded if at least one more specific route of the aggregation exists in the BGP routing table. Perform one of the following tasks to aggregate routes within BGP:

- Redistributing a Static Aggregate Route into BGP, page 67
- Configuring Conditional Aggregate Routes Using BGP, page 69
- Suppressing and Uns suppressing Advertising Aggregated Routes Using BGP, page 70
- Suppressing Inactive Route Advertisement Using BGP, page 72
- Conditionally Advertising BGP Routes, page 74

**Redistributing a Static Aggregate Route into BGP**

Use this task to redistribute a static aggregate route into BGP. A static aggregate route is configured and then redistributed into the BGP routing table. The static route must be configured to point to interface null 0 and the prefix should be a superset of known BGP routes. When a router receives a BGP packet it will use the more specific BGP routes. If the route is not found in the BGP routing table, then the packet will be forwarded to null 0 and discarded.
### SUMMARY STEPS

1. enable
2. configure terminal
3. ip route prefix mask \( [ip-address | interface-type interface-number [ip-address]] \) \([distance] [name] [permanent | track number] [tag tag]\)
4. router bgp autonomous-system-number
5. redistribute static
6. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| Example: | |
| Router> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example: | |
| Router# configure terminal | |
| **Step 3** ip route prefix mask \( [ip-address | interface-type interface-number [ip-address]] \) \([distance] [name] [permanent | track number] [tag tag]\) | Creates a static route. |
| Example: | |
| Router(config)# ip route 172.0.0.0 255.0.0.0 null 0 | |
| **Step 4** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| Example: | |
| Router(config)# router bgp 45000 | |
| **Step 5** redistribute static | Redistributes routes into the BGP routing table. |
| Example: | |
| Router(config-router)# redistribute static | |
Configuring Conditional Aggregate Routes Using BGP

Use this task to create an aggregate route entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route is advertised as originating from your autonomous system. For more information, see the GUID-51A47AB9-0A1B-4BF6-BEF3-AF5AE50661F6.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. aggregate-address address mask [as-set]
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th>aggregate-address address mask [as-set]</th>
</tr>
</thead>
</table>

**Purpose**

- Creates an aggregate entry in a BGP routing table.
- A specified route must exist in the BGP table.
- Use the `aggregate-address` command with no keywords to create an aggregate entry if any more-specific BGP routes are available that fall in the specified range.
- Use the `as-set` keyword to specify that the path advertised for this route is an AS-SET. Do not use the `as-set` keyword when aggregating many paths because this route is withdrawn and updated every time the reachability information for the aggregated route changes.

**Example:**

```
Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set
```

**Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>end</th>
</tr>
</thead>
</table>

**Purpose**

- Exits router configuration mode and enters privileged EXEC mode.

**Example:**

```
Router(config-router)# end
```

---

### Suppressing and Unsuppressing Advertising Aggregated Routes Using BGP

Use this task to create an aggregate route, suppress the advertisement of routes using BGP, and subsequently unsuppress the advertisement of routes. Routes that are suppressed are not advertised to any neighbors, but it is possible to unsuppress routes that were previously suppressed to specific neighbors.

#### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. Do one of the following:
   - aggregate-address address mask [summary-only]
   - aggregate-address address mask [suppress-map map-name]
6. neighbor {ip-address | peer-group-name} unsuppress-map map-name
7. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |

**Example:**  
Router> enable

<table>
<thead>
<tr>
<th><strong>Step 2</strong> configure terminal</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
</table>

**Example:**  
Router# configure terminal

<table>
<thead>
<tr>
<th><strong>Step 3</strong> router bgp autonomous-system-number</th>
<th>Enters router configuration mode for the specified routing process.</th>
</tr>
</thead>
</table>

**Example:**  
Router(config)# router bgp 45000

<table>
<thead>
<tr>
<th><strong>Step 4</strong> neighbor ip-address remote-as autonomous-system-number</th>
<th>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</th>
</tr>
</thead>
</table>

**Example:**  
Router(config-router)# neighbor 192.168.1.2 remote-as 40000

| **Step 5** Do one of the following: | Creates an aggregate route.  
• aggregate-address address mask  
  [summary-only]  
  •  
  • aggregate-address address mask [suppress-map map-name]  

**Example:**  
Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only

**Example:**  
Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 suppress-map map1

**Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*. |
### Command or Action

| Step 6 | **neighbor** \{ip-address | peer-group-name\} unsuppress-map map-name |
|--------|--------------------------------------------------|
| **Purpose** | (Optional) Selectively advertises routes previously suppressed by the aggregate-address command. |

**Example:**

```
Router(config-router)# neighbor 192.168.1.2 unsuppress map1
```

### Step 7 end

<table>
<thead>
<tr>
<th><strong>Purpose</strong></th>
<th>Exits router configuration mode and enters privileged EXEC mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# end</td>
</tr>
</tbody>
</table>

## Suppressing Inactive Route Advertisement Using BGP

Perform this task to suppress the advertisement of inactive routes by BGP. The `bgp suppress-inactive` command configures BGP to not advertise inactive routes to any BGP peer. A BGP routing process can advertise routes that are not installed in the RIB to BGP peers by default. A route that is not installed into the RIB is an inactive route. Inactive route advertisement can occur, for example, when routes are advertised through common route aggregation.

Inactive route advertisements can be suppressed to provide more consistent data forwarding. This feature can be configured on a per IPv4 address family basis. For example, when specifying the maximum number of routes that can be configured in a VRF with the `maximum routes` global configuration command, you also suppress inactive route advertisement to prevent inactive routes from being accepted into the VRF after route limit has been exceeded.

This task assumes that BGP is enabled and that peering has been established.

**Note**

Inactive route suppression can be configured only under the IPv4 address family or under a default IPv4 general session.

### SUMMARY STEPS

1. *enable*
2. *configure terminal*
3. *router bgp as-number*
4. *address-family* \{ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]\}
5. *bgp suppress-inactive*
6. *end*
7. *show ip bgp rib-failure*
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| **Example:** | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | Router# configure terminal |
| **Step 3** router bgp as-number | Enters router configuration mode and creates a BGP routing process. |
| **Example:** | Router(config)# router bgp 45000 |
| **Step 4** address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]} | Enter address family configuration mode to configure BGP peers to accept address family specific configurations.  
• The example creates an IPv4 unicast address family session. |
| **Example:** | Router(config-router)# address-family ipv4 unicast |
| **Step 5** bgp suppress-inactive | Suppresses BGP advertising of inactive routes.  
• BGP advertises inactive routes by default.  
• Entering the no form of this command reenables the advertisement of inactive routes. |
| **Example:** | Router(config-router-af)# bgp suppress-inactive |
| **Step 6** end | Exits address family configuration mode and enters privileged EXEC mode. |
| **Example:** | Router(config-router-af)# end |
| **Step 7** show ip bgp rib-failure | (Optional) Displays BGP routes that are not installed in the RIB. |
| **Example:** | Router# show ip bgp rib-failure |
Examples

The following example shows output from the `show ip bgp rib-failure` command displaying routes that are not installed in the RIB. The output shows that the displayed routes were not installed because a route or routes with a better administrative distance already exist in the RIB.

```plaintext
Router# show ip bgp rib-failure

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>RIB-failure</th>
<th>RIB-NH Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.15.0/24</td>
<td>10.1.35.5</td>
<td>Higher admin distance</td>
<td>n/a</td>
</tr>
<tr>
<td>10.1.16.0/24</td>
<td>10.1.15.1</td>
<td>Higher admin distance</td>
<td>n/a</td>
</tr>
</tbody>
</table>
```

Conditionally Advertising BGP Routes

Perform this task to conditionally advertise selected BGP routes. The routes or prefixes that will be conditionally advertised are defined in two route maps: an `advertise` map and either an `exist` map or `nonexist` map. The route map associated with the `exist` map or `nonexist` map specifies the prefix that the BGP speaker will track. The route map associated with the `advertise` map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

- If a prefix is found to be present in the `exist` map by the BGP speaker, then the prefix specified by the `advertise` map is advertised.
- If a prefix is found not to be present in the `nonexist` map by the BGP speaker, then the prefix specified by the `advertise` map is advertised.

If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised need to exist in the BGP routing table for conditional advertisement to occur. These routes are referenced from an access list or an IP prefix list.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `neighbor {ip-address | peer-group-name} remote-as autonomous-system-number`
5. `neighbor ip-address advertise-map map-name {exist-map map-name | non-exist-map map-name}`
6. `exit`
7. `route-map map-tag [permit|deny][sequence-number]`
8. `match ip address {access-list-number [access-list-number... | access-list-name...] | access-list-name [access-list-number... | access-list-name] | prefix-list prefix-list-name [prefix-list-name...]}`
9. `route-map map-tag [permit|deny][sequence-number]`
10. `match ip address {access-list-number [access-list-number... | access-list-name...] | access-list-name [access-list-number... | access-list-name] | prefix-list prefix-list-name [prefix-list-name...]}`
11. `exit`
12. `access-list access-list-number {deny | permit} source [source-wildcard] [log]`
13. `access-list access-list-number {deny | permit} source [source-wildcard] [log]`
14. `exit`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor ip-address advertise-map map-name</td>
<td>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</td>
</tr>
<tr>
<td>{exist-map map-name</td>
<td>non-exist-map map-name}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.1.2 advertise-map map1 exist-map map2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 7</strong> route-map map-tag [permit</td>
<td>deny][sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In this example, a route map named map1 is created.</td>
</tr>
<tr>
<td>Router(config)# route-map map1 permit 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> match ip address {access-list-number [access-list-number...</td>
<td>access-list-name... ]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In this example, the route map is configured to match a prefix permitted by access list 1.</td>
</tr>
<tr>
<td>Router(config-route-map)# match ip address 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> route-map map-tag [permit</td>
<td>deny][sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In this example, a route map named map2 is created.</td>
</tr>
<tr>
<td>Router(config)# route-map map2 permit 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> match ip address {access-list-number [access-list-number...</td>
<td>access-list-name... ]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In this example, the route map is configured to match a prefix permitted by access list 2.</td>
</tr>
<tr>
<td>Router(config-route-map)# match ip address 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> access-list access-list-number {deny</td>
<td>permit} source [source-wildcard] [log]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In this example, access list 1 permits advertising of the 172.17.0.0 prefix, depending on other conditions set by the neighbor advertise-map command.</td>
</tr>
<tr>
<td>Router(config)# access-list 1 permit 172.17.0.0</td>
<td></td>
</tr>
</tbody>
</table>
### Originating BGP Routes

Route aggregation is useful to minimize the size of the BGP table but there are situations when you want to add more specific prefixes to the BGP table. Route aggregation can hide more specific routes. Using the `network` command as shown in the "Configuring a BGP Routing Process" section originates routes, and the following optional tasks originate BGP routes for the BGP table for different situations.

- Advertising a Default Route Using BGP, page 77
- Conditionally Injecting BGP Routes, page 79
- Originating BGP Routes Using Backdoor Routes, page 83

### Advertising a Default Route Using BGP

Perform this task to advertise a default route to BGP peers. The default route is locally originated. A default route can be useful to simplify configuration or to prevent the router from using too many system resources. If the router is peered with an Internet service provider (ISP), the ISP will carry full routing tables, so configuring a default route into the ISP network saves resources at the local router.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. `ip prefix-list list-name [seq seq-value] (deny network / length) permit network / length) [ge ge-value] [le le-value]`
4. `route-map map-tag [permit|deny][sequence-number]`
5. `match ip address {access-list-number [access-list-number... | access-list-name...] | access-list-name [access-list-number... | access-list-name] | prefix-list prefix-list-name [prefix-list-name...]}`
6. **exit**
7. `router bgp autonomous-system-number`
8. `neighbor {ip-address | peer-group-name} [default-originate[route-map map-name]`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip prefix-list list-name [seq seq-value]</td>
<td>Configures an IP prefix list.</td>
</tr>
<tr>
<td>[deny network [length]]</td>
<td>In this example, prefix list DEFAULT permits advertising of the 10.1.1.0/24. prefix depending on a match set by the match ip address command.</td>
</tr>
<tr>
<td>[permit network [length]]</td>
<td></td>
</tr>
<tr>
<td>[ge ge-value]</td>
<td></td>
</tr>
<tr>
<td>[le le-value]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip prefix-list DEFAULT permit 10.1.1.0/24</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> route-map map-tag [permit] [deny][sequence-number]</td>
<td>Configures a route map and enters route map configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>In this example, a route map named ROUTE is created.</td>
</tr>
<tr>
<td>Router(config)# route-map ROUTE</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> match ip address {access-list-number [access-list-number... [access-list-name...]</td>
<td>Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.</td>
</tr>
<tr>
<td>] access-list-name [access-list-number... [access-list-name]</td>
<td>In this example, the route map is configured to match a prefix permitted by prefix list DEFAULT.</td>
</tr>
<tr>
<td>[prefix-list prefix-list-name [prefix-list-name...]]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# match ip address prefix-list DEFAULT</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command/Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# router bgp 40000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>`neighbor {ip-address</td>
<td>peer-group-name} default-originate[route-map map-name]`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router)# neighbor 192.168.3.2 default-originate</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>end</code></td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router)# end</td>
<td></td>
</tr>
</tbody>
</table>

#### Troubleshooting Tips

- Troubleshooting Tips, page 79

### Troubleshooting Tips

Use the `show ip route` command on the receiving BGP peer (not on the local router) to verify that the default route has been set. In the output, verify that a line similar to the following showing the default route 0.0.0.0 is present:

```
B*  0.0.0.0/0 [20/0] via 192.168.1.2, 00:03:10
```

### Conditionally Injecting BGP Routes

Use this task to inject more specific prefixes into a BGP routing table over less specific prefixes that were selected through normal route aggregation. These more specific prefixes can be used to provide a finer granularity of traffic engineering or administrative control than is possible with aggregated routes. For more information, see the "Conditional BGP Route Injection" section.

This task assumes that the IGP is already configured for the BGP peers.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. bgp inject-map inject-map-name exist-map exist-map-name [copy-attributes]
5. exit
6. route-map map-tag [permit|deny] [sequence-number]
7. match ip address [access-list-number | access-list-name | prefix-list prefix-list-name]
8. match ip route-source [access-list-number | access-list-name | prefix-list prefix-list-name]
9. exit
10. route-map map-tag [permit|deny] [sequence-number]
11. set ip address [access-list-number | access-list-name | prefix-list prefix-list-name]
12. set community [community-number | well-known-community | none]
13. exit
14. ip prefix-list list-name [seq seq-value] [deny network length | permit network length] [ge ge-value]
15. Repeat Step 14 for every prefix list to be created.
16. exit
17. show ip bgp injected-paths

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config) &gt; configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 40000</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 4** `bgp inject-map inject-map-name exist-map exist-map-name [copy-attributes]` | Specifies the inject map and the exist map for conditional route injection.  
  • Use the `copy-attributes` keyword to specify that the injected route inherit the attributes of the aggregate route. |
| **Example:**  
  `Router(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH` | |
| **Step 5** `exit` | Exits router configuration mode and enters global configuration mode. |
| **Example:**  
  `Router(config-router)# exit` | |
| **Step 6** `route-map map-tag [permit|deny][sequence-number]` | Configures a route map and enters route map configuration mode. |
| **Example:**  
  `Router(config)# route-map LEARNED_PATH permit 10` | |
| **Step 7** `match ip address {access-list-number [access-list-name...] | access-list-name [access-list-number...] | access-list-name | prefix-list prefix-list-name [prefix-list-name...]}` | Specifies the aggregate route to which a more specific route will be injected.  
  • In this example, the prefix list named SOURCE is used to redistribute the source of the route. |
| **Example:**  
  `Router(config-route-map)# match ip address prefix-list SOURCE` | |
| **Step 8** `match ip route-source {access-list-number | access-list-name} [access-list-number...] access-list-name...}]` | Specifies the match conditions for redistributing the source of the route.  
  • In this example, the prefix list named ROUTE_SOURCE is used to redistribute the source of the route.  
  **Note** The route source is the neighbor address that is configured with the `neighbor remote-as` command.  
  The tracked prefix must come from this neighbor in order for conditional route injection to occur. |
| **Example:**  
  `Router(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE` | |
| **Step 9** `exit` | Exits route map configuration mode and enters global configuration mode. |
| **Example:**  
  `Router(config-route-map)# exit` | |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong> route-map map-tag [permit</td>
<td>deny][sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# route-map ORIGINATE permit 10</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 11** set ip address {access-list-number [access-list-number... | Specifies the routes to be injected.  
  - In this example, the prefix list named originated_routes is used to redistribute the source of the route. |
|   access-list-name...] [access-list-name [access-list-number...] access-list-name] |   |
|   [prefix-list prefix-list-name [prefix-list-name...]] |   |
| **Example:** Router(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES | |
| **Step 12** set community [community-number [additive] [well-known-community] | Sets the BGP community attribute of the injected route. |
|   known-community] |   |
|   [none] } |   |
| **Example:** Router(config-route-map)# set community 14616:555 additive | |
| **Step 13** exit | Exits route map configuration mode and enters global configuration mode. |
| **Example:** Router(config-route-map)# exit | |
| **Step 14** ip prefix-list list-name [seq seq-value] [deny network | Configures a prefix list.  
  - In this example, the prefix list named SOURCE is configured to permit routes from network 10.1.1.0/24. |
|   length | permit network | ge ge-value] [le le-value] |   |
| **Example:** Router(config)# ip prefix-list SOURCE permit 10.1.1.0/24 | |
| **Step 15** Repeat Step 14 for every prefix list to be created. | -- |
| **Step 16** exit | Exits global configuration mode and returns to privileged EXEC mode. |
| **Example:** Router(config)# exit | |
**Command or Action** | **Purpose**  
---|---  
**Step 17** show ip bgp injected-paths | (Optional) Displays information about injected paths.  

**Example:**  
Router# show ip bgp injected-paths

**Examples**  
The following sample output is similar to the output that will be displayed when the `show ip bgp injected-paths` command is entered:

```plaintext
Router# show ip bgp injected-paths
BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes:i - IGP, e - EGP, ? - incomplete
Network  Next Hop  Metric LocPrf Weight Path
*> 172.16.0.0  10.0.0.2  0  
*> 172.17.0.0/16  10.0.0.2  0  
```

- Troubleshooting Tips, page 83

**Troubleshooting Tips**  
BGP conditional route injection is based on the injection of a more specific prefix into the BGP routing table when a less specific prefix is present. If conditional route injection is not working properly, verify the following:

- If conditional route injection is configured but does not occur, verify the existence of the aggregate prefix in the BGP routing table. The existence (or not) of the tracked prefix in the BGP routing table can be verified with the `show ip bgp` command.
- If the aggregate prefix exists but conditional route injection does not occur, verify that the aggregate prefix is being received from the correct neighbor and the prefix list identifying that neighbor is a /32 match.
- Verify the injection (or not) of the more specific prefix using the `show ip bgp injected-paths` command.
- Verify that the prefix that is being injected is not outside of the scope of the aggregate prefix.
- Ensure that the inject route map is configured with the `set ip address` command and not the `match ip address` command.

**Originating BGP Routes Using Backdoor Routes**  
Use this task to indicate to border routers which networks are reachable using a backdoor route. A backdoor network is treated the same as a local network except that it is not advertised. For more information see the BGP Backdoor Routes, page 31.

This task assumes that the IGP--EIGRP in this example--is already configured for the BGP peers. The configuration is done at Router B in the figure above (in the "BGP Backdoor Routes" section) and the BGP peer is Router D.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `neighbor ip-address remote-as autonomous-system-number`
5. `network ip-address backdoor`
6. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `enable` | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Example:** | Router> enable |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:** | Router# configure terminal |
| **Step 3** `router bgp autonomous-system-number` | Enters router configuration mode for the specified routing process. |
| **Example:** | Router(config)# router bgp 45000 |
| **Step 4** `neighbor ip-address remote-as autonomous-system-number` | Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local router.  
  • In this example, the peer is an internal peer as the autonomous system number specified for the peer is the same number specified in Step 3. |
| **Example:** | Router(config-router)# neighbor 172.22.1.2 remote-as 45000 |
| **Step 5** `network ip-address backdoor` | Indicates a network that is reachable through a backdoor route. |
| **Example:** | Router(config-router)# network 172.21.1.0 backdoor |
### Configuring a BGP Peer Group

This task explains how to configure a BGP peer group. Often, in a BGP speaker, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into peer groups to simplify configuration and, more importantly, to make updating more efficient. When you have many peers, this approach is highly recommended.

The three steps to configure a BGP peer group, described in the following task, are as follows:

- Creating the peer group
- Assigning options to the peer group
- Making neighbors members of the peer group

You can disable a BGP peer or peer group without removing all the configuration information using the `neighbor shutdown` router configuration command.

#### Note

By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types.

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `neighbor peer-group-name peer-group`
5. `neighbor ip-address remote-as autonomous-system-number`
6. `neighbor ip-address peer-group peer-group-name`
7. `address-family ipv4 [unicast | multicast] vrf vrf-name`
8. `neighbor peer-group-name activate`
9. `neighbor ip-address peer-group peer-group-name`
10. `end`
## Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. | Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. | Router(config)# router bgp 40000 |
| **Step 4** neighbor peer-group-name peer-group | Creates a BGP peer group. | Router(config-router)# neighbor fingroup peer-group |
| **Step 5** neighbor ip-address remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local router. | Router(config-router)# neighbor 192.168.1.1 remote-as 45000 |
| **Step 6** neighbor ip-address peer-group peer-group-name | Assigns the IP address of a BGP neighbor to a peer group. | Router(config-router)# neighbor 192.168.1.1 peer-group fingroup |
### Configuring Peer Session Templates

The following tasks create and configure a peer session template:

- Configuring a Basic Peer Session Template, page 87
- Configuring Peer Session Template Inheritance with the inherit peer-session Command, page 90
- Configuring Peer Session Template Inheritance with the neighbor inherit peer-session Command, page 92

#### Configuring a Basic Peer Session Template

Perform this task to create a basic peer session template with general BGP routing session commands that can be applied to many neighbors using one of the next two tasks.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> address-family ipv4 [unicast</td>
<td>multcast] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# address-family ipv4 multicast</td>
</tr>
<tr>
<td><strong>Step 8</strong> neighbor peer-group-name activate</td>
<td>Enables the neighbor to exchange prefixes for the IPv4 address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# neighbor fingroup activate</td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor ip-address peer-group peer-group-name</td>
<td>Assigns the IP address of a BGP neighbor to a peer group.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# neighbor 192.168.1.1 peer-group fingroup</td>
</tr>
<tr>
<td><strong>Step 10</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# end</td>
</tr>
</tbody>
</table>
Note

The commands in Step 5 and 6 are optional and could be replaced with any supported general session commands.

Note

The following restrictions apply to the peer session templates:

- A peer session template can directly inherit only one session template, and each inherited session template can also contain one indirectly inherited session template. So, a neighbor or neighbor group can be configured with only one directly applied peer session template and seven additional indirectly inherited peer session templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. template peer-session session-template-name
5. remote-as autonomous-system-number
6. timers keepalive-interval hold-time
7. end
8. show ip bgp template peer-session [session-template-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enter your password if prompted.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> template peer-session session-template-name</td>
<td>Enters session-template configuration mode and creates a peer session template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# template peer-session INTERNAL-BGP</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> remote-as autonomous-system-number</td>
<td>(Optional) Configures peering with a remote neighbor in the specified autonomous system.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Note Any supported general session command can be used here. For a list of the supported commands, see the &quot;Restrictions&quot; section.</td>
</tr>
<tr>
<td>Router(config-router-stmp)# remote-as 202</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> timers keepalive-interval hold-time</td>
<td>(Optional) Configures BGP keepalive and hold timers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Note Any supported general session command can be used here. For a list of the supported commands, see the &quot;Restrictions&quot; section.</td>
</tr>
<tr>
<td>Router(config-router-stmp)# timers 30 300</td>
<td>• The hold time must be at least twice the keepalive time.</td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits session-template configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show ip bgp template peer-session [session-template-name]</td>
<td>Displays locally configured peer session templates.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The output can be filtered to display a single peer policy template with the session-template-name argument. This command also supports all standard output modifiers.</td>
</tr>
<tr>
<td>Router# show ip bgp template peer-session</td>
<td></td>
</tr>
</tbody>
</table>

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.
Configuring Peer Session Template Inheritance with the `inherit peer-session` Command

This task configures peer session template inheritance with the `inherit peer-session` command. It creates and configures a peer session template and allows it to inherit a configuration from another peer session template.

**Note**
The commands in Steps 5 and 6 are optional and could be replaced with any supported general session commands.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `template peer-session session-template-name`
5. `description text-string`
6. `update-source interface-type interface-number`
7. `inherit peer-session session-template-name`
8. `end`
9. `show ip bgp template peer-session [session-template-name]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 101</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong> template peer-session <em>session-template-name</em></td>
<td>Enter session-template configuration mode and creates a peer session template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# template peer-session CORE1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> description text-string</td>
<td>(Optional) Configures a description.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-stmp)# description CORE-123</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> update-source interface-type interface-number</td>
<td>(Optional) Configures a router to select a specific source or interface to receive routing table updates.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-stmp)# update-source loopback 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> inherit peer-session <em>session-template-name</em></td>
<td>Configures this peer session template to inherit the configuration of another peer session template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-stmp)# inherit peer-session INTERNAL-BGP</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits session-template configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> show ip bgp template peer-session [session-template-name]</td>
<td>Displays locally configured peer session templates.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp template peer-session</td>
<td></td>
</tr>
</tbody>
</table>

- What to Do Next, page 92
What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the `inherit peer-session` or `neighbor inherit peer-session` command.

Configuring Peer Session Template Inheritance with the `neighbor inherit peer-session` Command

This task configures a router to send a peer session template to a neighbor to inherit the configuration from the specified peer session template with the `neighbor inherit peer-session` command. Use the following steps to send a peer session template configuration to a neighbor to inherit:

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `neighbor ip-address remote-as autonomous-system-number`
5. `neighbor ip-address inherit peer-session session-template-name`
6. `end`
7. `show ip bgp template peer-session [session-template-name]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 101</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 4
- **neighbor ip-address remote-as autonomous-system-number**
  
  **Example:**
  
  ```
  Router(config-router)# neighbor 172.16.0.1 remote-as 202
  ```

  **Purpose:** Configures a peering session with the specified neighbor.
  
  - The explicit `remote-as` statement is required for the neighbor inherit statement in Step 5 to work. If a peering is not configured, the specified neighbor in Step 5 will not accept the session template.

#### Step 5
- **neighbor ip-address inherit peer-session session-template-name**
  
  **Example:**
  
  ```
  Router(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1
  ```

  **Purpose:** Sends a peer session template to a neighbor so that the neighbor can inherit the configuration.
  
  - The example configures a router to send the peer session template named CORE1 to the 172.16.0.1 neighbor to inherit. This template can be applied to a neighbor, and if another peer session template is indirectly inherited in CORE1, the indirectly inherited configuration will also be applied. No additional peer session templates can be directly applied. However, the directly inherited template can also inherit up to seven additional indirectly inherited peer session templates.

#### Step 6
- **end**
  
  **Example:**
  
  ```
  Router(config-router)# end
  ```

  **Purpose:** Exits router configuration mode and enters privileged EXEC mode.

#### Step 7
- **show ip bgp template peer-session [session-template-name]**
  
  **Example:**
  
  ```
  Router# show ip bgp template peer-session
  ```

  **Purpose:** Displays locally configured peer session templates.
  
  - The output can be filtered to display a single peer policy template with the optional `session-template-name` argument. This command also supports all standard output modifiers.

---

**What to Do Next**

To create a peer policy template, go to the Configuring Peer Policy Templates, page 93.

### Configuring Peer Policy Templates

The following tasks create and configure a peer policy template:

- Configuring Basic Peer Policy Templates, page 94
- Configuring Peer Policy Template Inheritance with the inherit peer-policy Command, page 96
- Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command, page 98
Configuring Basic Peer Policy Templates

Perform this task to create a basic peer policy template with BGP policy configuration commands that can be applied to many neighbors using one of the next two tasks.

**Note**
The commands in Steps 5 through 7 are optional and could be replaced with any supported BGP policy configuration commands.

**Note**
The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `template peer-policy policy-template-name`
5. `maximum-prefix prefix-limit [threshold] [restart restart-interval | warning-only]`
6. `weight weight-value`
7. `prefix-list prefix-list-name {in | out}`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> template peer-policy policy-template-name</td>
<td>Enters policy-template configuration mode and creates a peer policy template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# template peer-policy GLOBAL</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> maximum-prefix prefix-limit [threshold] [restart restart-interval</td>
<td>(Optional) Configures the maximum number of prefixes that a neighbor will accept from this peer.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-ptmp)# maximum-prefix 10000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> weight weight-value</td>
<td>(Optional) Sets the default weight for routes that are sent from this neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong> Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the Peer Policy Templates, page 35.</td>
</tr>
<tr>
<td>Router(config-router-ptmp)# weight 300</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> prefix-list prefix-list-name {in</td>
<td>out}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong> Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the Peer Policy Templates, page 35.</td>
</tr>
<tr>
<td>Router(config-router-ptmp)# prefix-list NO-MARKETING in</td>
<td>• The prefix list in the example filters inbound internal addresses.</td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits policy-template configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-ptmp)# end</td>
<td></td>
</tr>
</tbody>
</table>

**What to Do Next**

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template. For more details about peer policy inheritance see the
"Configuring Peer Policy Template Inheritance with the inherit peer-policy Command" section or the "Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command" section.

**Configuring Peer Policy Template Inheritance with the inherit peer-policy Command**

This task configures peer policy template inheritance using the `inherit peer-policy` command. It creates and configure a peer policy template and allows it to inherit a configuration from another peer policy template.

When BGP neighbors use inherited peer templates, it can be difficult to determine which policies are associated with a specific template. In Cisco IOS XE Release 2.1 and later releases, the `detail` keyword was added to the `show ip bgp template peer-policy` command to display the detailed configuration of local and inherited policies associated with a specific template.

**Note**
The commands in Steps 5 and 6 are optional and could be replaced with any supported BGP policy configuration commands.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. template peer-policy policy-template-name
5. route-map map-name {in out}
6. inherit peer-policy policy-template-name sequence-number
7. end
8. show ip bgp template peer-policy [policy-template-name[detail]]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> template peer-policy policy-template-name</td>
<td>Enter policy-template configuration mode and creates a peer policy template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# template peer-policy NETWORK1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> route-map map-name {in</td>
<td>out}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong> Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the GUID-578E3188-1BFE-453D-82DC-9F147308B683.</td>
</tr>
<tr>
<td>Router(config-router-ptmp)# route-map ROUTE in</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> inherit peer-policy policy-template-name sequence-number</td>
<td>(Optional) Configures the peer policy template to inherit the configuration of another peer policy template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The <em>sequence-number</em> argument sets the order in which the peer policy template is evaluated. Like a route map sequence number, the lowest sequence number is evaluated first.</td>
</tr>
<tr>
<td>Router(config-router-ptmp)# inherit peer-policy GLOBAL 10</td>
<td>• The example configures this peer policy template to inherit the configuration from GLOBAL. If the template created in these steps is applied to a neighbor, the configuration GLOBAL will also be inherited and applied indirectly. Up to six additional peer policy templates can be indirectly inherited from GLOBAL for a total of eight directly applied and indirectly inherited peer policy templates.</td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits policy-template configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-ptmp)# end</td>
<td></td>
</tr>
</tbody>
</table>
### Examples

The following sample output of the `show ip bgp template peer-policy` command with the `detail` keyword displays details of the policy named NETWORK1. The output in this example shows that the GLOBAL template was inherited. Details of route map and prefix list configurations are also displayed.

```
Router# show ip bgp template peer-policy NETWORK1 detail
Template:NETWORK1, index:2.
Local policies:0x1, Inherited polices:0x80840
This template inherits:
  GLOBAL, index:1, seq_no:10, flags:0x1
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  weight 300
  maximum-prefix 10000
Template:NETWORK1 <detail>
Locally configured policies:
  route-map ROUTE in
  route-map ROUTE, permit, sequence 10
Match clauses:
  ip address prefix-lists: DEFAULT
  ip prefix-list DEFAULT: 1 entries
    seq 5 permit 10.1.1.0/24
Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Inherited policies:
  prefix-list NO-MARKETING in
  ip prefix-list NO-MARKETING: 1 entries
    seq 5 deny 10.2.2.0/24
```

### Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command

This task configures a router to send a peer policy template to a neighbor to inherit using the `neighbor inherit peer-policy` command. Perform the following steps to send a peer policy template configuration to a neighbor to inherit.

When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. In Cisco IOS XE Release 2.1 and later releases, the `policy` and `detail` keywords were added to the `show ip bgp neighbors` command to display the inherited policies and policies configured directly on the specified neighbor.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. address-family ipv4 [multicast | unicast | vrf vrf-name]
6. neighbor ip-address inherit peer-policy policy-template-name
7. end
8. show ip bgp neighbors [ip-address[policy [detail]]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Configures a peering session with the specified neighbor.</td>
</tr>
<tr>
<td>Example:</td>
<td>• The explicit remote-as statement is required for the neighbor inherit statement in Step 6 to work. If a peering is not configured, the specified neighbor in Step 6 will not accept the session template.</td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family ipv4 [multicast</td>
<td>unicast</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 6** neighbor ip-address inherit peer-policy policy-template-name | Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.  
  - The example configures a router to send the peer policy template named GLOBAL to the 192.168.1.2 neighbor to inherit. This template can be applied to a neighbor, and if another peer policy template is indirectly inherited from GLOBAL, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from GLOBAL. |
| **Example:** | Router(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL |
| **Step 7** end | Exits address family configuration mode and returns to privileged EXEC mode. |
| **Example:** | Router(config-router-af)# end |
| **Step 8** show ip bgp neighbors [ip-address][policy [detail]] | Displays locally configured peer policy templates.  
  - The output can be filtered to display a single peer policy template with the policy-template-name argument. This command also supports all standard output modifiers.  
  - Use the policy keyword to display the policies applied to this neighbor per address family.  
  - Use the detail keyword to display detailed policy information. |
| **Example:** | Router# show ip bgp neighbors 192.168.1.2 policy |

**Note** Only the syntax required for this task is shown. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

### Examples

The following sample output shows the policies applied to the neighbor at 192.168.1.2. The output displays both inherited policies and policies configured on the neighbor device. Inherited policies are policies that the neighbor inherits from a peer-group or a peer-policy template.

```
Router# show ip bgp neighbors 192.168.1.2 policy  
Neighbor: 192.168.1.2, Address-Family: IPv4 Unicast  
Locally configured policies:  
  route-map ROUTE in  
Inherited policies:  
  prefix-list NO-MARKETING in  
  route-map ROUTE in  
  weight 300  
  maximum-prefix 10000
```

### Monitoring and Maintaining BGP Dynamic Update Groups

Use this task to clear and display information about the processing of dynamic BGP update groups. The performance of BGP update message generation is improved with the use of BGP update groups. With the configuration of the BGP peer templates and the support of the dynamic BGP update groups, the network operator no longer needs to configure peer groups in BGP and can benefit from improved configuration flexibility and system performance. For more information about using BGP peer templates, see the *Configuring Peer Session Templates*, page 87 and the *Configuring Peer Policy Templates*, page 93.
### SUMMARY STEPS

1. enable
2. `clear ip bgp update-group [index-group] ip-address`
3. `show ip bgp replication [index-group] ip-address`
4. `show ip bgp update-group [index-group] ip-address` [summary]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| **Example:** | |
| Router> enable | |
| **Step 2** clear ip bgp update-group [index-group] ip-address | Clears BGP update group membership and recalculate BGP update groups:  
• In the example provided, the membership of neighbor 192.168.2.2 is cleared from an update group. |
| **Example:** | |
| Router# clear ip bgp update-group 192.168.2.2 | |
| **Step 3** show ip bgp replication [index-group] ip-address | Displays update replication statistics for BGP update groups. |
| **Example:** | |
| Router# show ip bgp replication | |
| **Step 4** show ip bgp update-group [index-group] ip-address [summary] | Displays information about BGP update groups. |
| **Example:** | |
| Router# show ip bgp update-group | |

• Troubleshooting Tips, page 101

### Troubleshooting Tips

Use the **debug ip bgp groups** command to display information about the processing of BGP update groups. Information can be displayed for all update groups, an individual update group, or a specific BGP neighbor. The output of this command can be very verbose. This command should not be deployed in a production network unless your are troubleshooting a problem.
Configuration Examples for a Basic BGP Network

- Example Configuring a BGP Process and Customizing Peers, page 102
- Example Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers, page 103
- Example Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number, page 105
- Example NLRI to AFI Configuration, page 106
- Examples Removing BGP Configuration Commands Using a Redistribution Example, page 108
- Examples BGP Soft Reset, page 109
- Examples Resetting BGP Peers Using 4-Byte Autonomous System Numbers, page 109
- Example Resetting and Displaying Basic BGP Information, page 110
- Examples Aggregating Prefixes Using BGP, page 111
- Example Configuring a BGP Peer Group, page 112
- Example Configuring Peer Session Templates, page 112
- Example Configuring Peer Policy Templates, page 113
- Examples Monitoring and Maintaining BGP Dynamic Update Peer-Groups, page 113

Example Configuring a BGP Process and Customizing Peers

The following example shows the configuration for Router B in the figure above (in the "Customizing a BGP Peer" section) with a BGP process configured with two neighbor peers (at Router A and at Router E) in separate autonomous systems. IPv4 unicast routes are exchanged with both peers and IPv4 multicast routes are exchanged with the BGP peer at Router E.

Router B

```
router bgp 45000
  bgp router-id 172.17.1.99
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
neighbor 192.168.3.2 description finance

  address-family ipv4
    neighbor 192.168.1.2 activate
    neighbor 192.168.3.2 activate
    no auto-summary
    no synchronization
    network 172.17.1.0 mask 255.255.255.0
    exit-address-family

  address-family ipv4 multicast
    neighbor 192.168.3.2 activate
    neighbor 192.168.3.2 advertisement-interval 25
    no auto-summary
    no synchronization
    network 172.17.1.0 mask 255.255.255.0
    exit-address-family
```
Example Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS XE Release 2.4 and Later Releases

The following example is available in Cisco IOS XE Release 2.4 and later releases, and shows the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using asplain notation. IPv4 unicast routes are exchanged with all peers.

Figure 10  BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format

Router A

```bash
router bgp 65536
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
gbg log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 65538
e!
address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```bash
router bgp 65538
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
gbg log-neighbor-changes
timers bgp 70 120
```
Asdot Default Format in Cisco IOS XE Release 2.3

The following example of the asdot format is available in Cisco IOS XE Release 2.3, and shows how to create the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

**Figure 11**  BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format

Router A

```
router bgp 1.0
```

Router B

```
neighbor 192.168.1.1 remote-as 65536
neighbor 192.168.3.2 remote-as 65550
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family
```

Router E

```
router bgp 65550
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family
```

```
Asdot Default Format in Cisco IOS XE Release 2.3

The following example of the asdot format is available in Cisco IOS XE Release 2.3, and shows how to create the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

**Figure 11**  BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format

```
10.1.1.1  Router A

AS 1.0

192.168.1.2 eBGP

192.168.3.1 eBGP

172.17.1.1

AS 1.2

192.168.3.1

192.168.3.2

AS 1.14

10.2.2.2

Router E

Router A

router bgp 1.0
```
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.1.1 activate
no auto-summary
no synchronization
network 10.1.1.0 mask 255.255.255.0
exit-address-family

Router B

router bgp 1.2
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 1.0
neighbor 192.168.3.2 remote-as 1.14
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family

Router E

router bgp 1.14
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family

Example Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number

Asplain Default Format in Cisco IOS XE Release 2.4 and Later Releases

The following example is available in Cisco IOS XE Release 2.4 and later releases, and shows how to create a VRF with a route-target that uses a 4-byte autonomous system number, 65537, and how to set the route target to extended community value 65537:100 for routes that are permitted by the route map.

ip vrf vpn_red
rd 64500:100
route-target both 65537:100
exit
route-map red_map permit 10
set extcommunity rt 65537:100
end

After the configuration is completed, use the `show route-map` command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 65537.

RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
      extended community RT:65537:100
Policy routing matches: 0 packets, 0 bytes

**Asdot Default Format in Cisco IOS XE Release 2.3**

The following example of the asdot default format is available in Cisco IOS XE Release 2.3, and shows how to create a VRF with a route-target that uses a 4-byte autonomous system number, 1.1, and how to set the route target to extended community value 1.1:100 for routes that are permitted by the route map.

```plaintext
ip vrf vpn_red
  rd 64500:100
  route-target both 1.1:100
  exit
route-map red_map permit 10
  set extcommunity rt 1.1:100
end
```

After the configuration is completed, use the `show route-map` command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 1.1.

RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
      extended community RT:1.1:100
Policy routing matches: 0 packets, 0 bytes

**Example NLRI to AFI Configuration**

The following example upgrades an existing router configuration file in the NLRI format to the AFI format and set the router CLI to use only commands in the AFI format:

```plaintext
router bgp 60000
  bgp upgrade-cli
```

The `show running-config` command can be used in privileged EXEC mode to verify that an existing router configuration file has been upgraded from the NLRI format to the AFI format. The following sections provide sample output from a router configuration file in the NLRI format, and the same router configuration file after it has been upgraded to the AFI format with the `bgp upgrade-cli` command in router configuration mode.

---

**Note**

After a router has been upgraded from the AFI format to the NLRI format with the `bgp upgrade-cli` command, NLRI commands will no longer be accessible or configurable.

**Router Configuration File in NLRI Format Before Upgrading**

The following sample output is from the `show running-config` command in privileged EXEC mode. The sample output shows a router configuration file, in the NLRI format, prior to upgrading to the AFI format:
with the `bgp upgrade-cli` command. The sample output is filtered to show only the affected portion of the router configuration.

```
Router# show running-config | begin bgp
router bgp 101
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.1.1.1 remote-as 505 nlri unicast multicast
  no auto-summary
  !
  ip default-gateway 10.4.9.1
  ip classless
  !
  !
  route-map REDISTRIBUTE-MULTICAST allow 10
  match ip address prefix-list MULTICAST-PREFIXES
  set nlri multicast
  !
  route-map MULTICAST-PREFIXES allow 10
  !
  route-map REDISTRIBUTE-UNICAST allow 20
  match ip address prefix-list UNICAST-PREFIXES
  set nlri unicast
  !
  !
  line con 0
  line aux 0
  line vty 0 4
  !
  password PASSWORD

login

! end
```

**Router Configuration File in AFI Format After Upgrading**

The following sample output shows the router configuration file after it has been upgraded to the AFI format. The sample output is filtered to show only the affected portion of the router configuration file.

```
Router# show running-config | begin bgp
router bgp 101
  bgp log-neighbor-changes
  neighbor 10.1.1.1 remote-as 505
  no auto-summary
  !
  address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family ipv4
  neighbor 10.1.1.1 activate
  no auto-summary
  no synchronization
  exit-address-family
  !
  ip default-gateway 10.4.9.1
  ip classless
  !
  !
  route-map REDISTRIBUTE-MULTICAST_mcast allow 10
  match ip address prefix-list MULTICAST-PREFIXES
  !
  route-map REDISTRIBUTE-MULTICAST allow 10
  match ip address prefix-list MULTICAST-PREFIXES
  !
  route-map MULTICAST-PREFIXES allow 10
  !
  route-map REDISTRIBUTE-UNICAST allow 20
```

**Configuring a Basic BGP Network**

**Configuration Examples for a Basic BGP Network**

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2
match ip address prefix-list UNICAST-PREFIXES
!
!
line con 0
line aux 0
line vty 0 4
password PASSWORD
login
!
end

Examples Removing BGP Configuration Commands Using a Redistribution Example

The following examples show both the CLI configuration to enable the redistribution of BGP routes into EIGRP using a route map, and the CLI configuration to remove the redistribution and route map. Some BGP configuration commands can affect other CLI commands and this example demonstrates how the removal of one command affects another command.

In the first configuration example, a route map is configured to match and set autonomous system numbers. BGP neighbors in three different autonomous systems are configured and activated. An EIGRP routing process is started, and the redistribution of BGP routes into EIGRP using the route map is configured.

**CLI to Enable BGP Route Redistribution Into EIGRP**

route-map bgp-to-eigrp permit 10
match tag 50000
set tag 65000
exit
router bgp 45000
  bgp log-neighbor-changes
  address-family ipv4
    neighbor 172.16.1.2 remote-as 45000
    neighbor 172.21.1.2 remote-as 45000
    neighbor 192.168.3.2 remote-as 40000
    neighbor 192.168.3.2 remote-as 50000
    neighbor 172.16.1.2 activate
    neighbor 172.21.1.2 activate
    neighbor 192.168.1.2 activate
    neighbor 192.168.3.2 activate
    network 172.17.1.0 mask 255.255.255.0
    exit-address-family
exit
router eigrp 100
redistribute bgp 45000 metric 10000 100 255 1 1500 route-map bgp-to-eigrp
no auto-summary
exit

In the second configuration example, both the `route-map` command and the `redistribute` command are disabled. If only the route-map command is removed, it does not automatically disable the redistribution. The redistribution will now occur without any matching or filtering. To remove the redistribution configuration, the `redistribute` command must also be disabled.

**CLI to Remove BGP Route Redistribution Into EIGRP**

configure terminal
  no route-map bgp-to-eigrp
  router eigrp 100
  no redistribute bgp 45000
end
Examples BGP Soft Reset

The following examples show two ways to reset the connection for BGP peer 192.168.1.1.

Dynamic Inbound Soft Reset Example

The following example shows the `clear ip bgp 192.168.1.1 soft in` EXEC command used to initiate a dynamic soft reconfiguration in the BGP peer 192.168.1.1. This command requires that the peer support the route refresh capability.

`clear ip bgp 192.168.1.1 soft in`

Inbound Soft Reset Using Stored Information Example

The following example shows how to enable inbound soft reconfiguration for the neighbor 192.168.1.1. All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is performed later, the stored information will be used to generate a new set of inbound updates.

```
router bgp 100
neighbor 192.168.1.1 remote-as 200
neighbor 192.168.1.1 soft-reconfiguration inbound
```

The following example clears the session with the neighbor 192.168.1.1:

`clear ip bgp 192.168.1.1 soft in`

Examples Resetting BGP Peers Using 4-Byte Autonomous System Numbers

The following examples show how to clear BGP peers belonging to an autonomous system that uses 4-byte autonomous system numbers. This example requires Cisco IOS XE Release 2.4 or a later release to be running on the router. The initial state of the BGP routing table is shown using the `show ip bgp` command, and peers in 4-byte autonomous systems 65536 and 65550 are displayed.

```
RouterB# show ip bgp
BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 65536  i
*> 10.2.2.0/24      192.168.3.2              0             0 65550  i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

The `clear ip bgp 65550` command is entered to remove all BGP peers in the 4-byte autonomous system 65550. The ADJCHANGE message shows that the BGP peer at 192.168.3.2 is being reset.

```
RouterB# clear ip bgp 65550
RouterB# *Nov 30 23:25:27.043: %BGP-5-ADJCHANGE: neighbor 192.168.3.2 Down User reset
```

The `show ip bgp` command is entered again, and only the peer in 4-byte autonomous systems 65536 is now displayed.

```
RouterB# show ip bgp
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
```

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2

109
Almost immediately the next ADJCHANGE message shows that the BGP peer at 192.168.3.2 (in the 4-byte autonomous system 65550) is now back up.

**Example Resetting and Displaying Basic BGP Information**

The following example shows how to reset and display basic BGP information.

The `clear ip bgp *` command clears and resets all the BGP neighbor sessions. Specific neighbors or all peers in an autonomous system can be cleared by using the `neighbor-address` and `autonomous-system-number` arguments. If no argument is specified, this command will clear and reset all BGP neighbor sessions.

**Note**

The `clear ip bgp *` command also clears all the internal BGP structures which makes it useful as a troubleshooting tool.

```
Router# clear ip bgp *
```

The `show ip bgp` command is used to display all the entries in the BGP routing table. The following example displays BGP routing table information for the 10.1.1.0 network:

```
Router# show ip bgp 10.1.1.0 255.255.255.0
BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
   Advertised to update-groups: 1
   40000
   192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The `show ip bgp neighbors` command is used to display information about the TCP and BGP connections to neighbors. The following example displays the routes that were advertised from Router B in the figure above (in the "Configuring a BGP Peer for the IPv4 VRF Address Family" section) to its BGP neighbor 192.168.3.2 on Router E:

```
Router# show ip bgp neighbors 192.168.3.2 advertised-routes
BGP table version is 3, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network     Next Hop     Metric LocPrf Weight Path
   * 10.1.1.0/24 192.168.1.2   0     0       40000 i
   * 172.17.1.0/24 0.0.0.0     0     0       32768 i
Total number of prefixes 2
```

The `show ip bgp paths` command is used to display all the BGP paths in the database. The following example displays BGP path information for Router B in the figure above (in the "Customizing a BGP Peer" section):

```
Router# show ip bgp paths
Address     Hash Refcount Metric Path
0x2FB5DB0   0       5     0 i
```
The `show ip bgp summary` command is used to display the status of all BGP connections. The following example displays BGP routing table information for Router B in the figure above (in the "Customizing a BGP Peer" section):

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 45000
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
4/2 BGP path/bestpath attribute entries using 496 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 882 total bytes of memory
BGP activity 14/10 prefixes, 16/12 paths, scan interval 60 secs
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
192.168.1.2 4 40000 667 672 3 0 00:03:49 1
192.168.3.2 4 50000 468 467 0 0 00:03:49 (NoNeg)
```

Examples Aggregating Prefixes Using BGP

The following examples show how you can use aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the BGP conditional aggregation routing feature.

In the following example, the `redistribute static` router configuration command is used to redistribute aggregate route 10.0.0.0:

```
ip route 10.0.0.0 255.0.0.0 null 0
!
router bgp 100
  redistribute static
```

The following configuration shows how to create an aggregate entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route will be advertised as coming from your autonomous system and has the atomic aggregate attribute set to show that information might be missing. (By default, atomic aggregate is set unless you use the `as-set` keyword in the `aggregate-address` router configuration command.)

```
router bgp 100
  aggregate-address 10.0.0.0 255.0.0.0
```

The following example shows how to create an aggregate entry using the same rules as in the previous example, but the path advertised for this route will be an AS-SET consisting of all elements contained in all paths that are being summarized:

```
router bgp 100
  aggregate-address 10.0.0.0 255.0.0.0 as-set
```

The following example shows how to create the aggregate route for 10.0.0.0 and also suppress advertisements of more specific routes to all neighbors:

```
router bgp 100
  aggregate-address 10.0.0.0 255.0.0.0 summary-only
```

The following example, starting in global configuration mode, configures BGP to not advertise inactive routes:

```
Router(config)# router bgp 50000
Router(config-router)# address-family ipv4 unicast
```
Example Configuring a BGP Peer Group

The following example shows how to use an address family to configure a peer group so that all members of the peer group are both unicast- and multicast-capable:

```plaintext
router bgp 45000
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
address-family ipv4 unicast
    neighbor mygroup peer-group
    neighbor 192.168.1.2 peer-group mygroup
    neighbor 192.168.3.2 peer-group mygroup
router bgp 45000
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
address-family ipv4 multicast
    neighbor mygroup peer-group
    neighbor 192.168.1.2 peer-group mygroup
    neighbor 192.168.3.2 peer-group mygroup
    neighbor 192.168.1.2 activate
    neighbor 192.168.3.2 activate
```

Example Configuring Peer Session Templates

The following example creates a peer session template named INTERNAL-BGP in session-template configuration mode:

```plaintext
router bgp 45000
template peer-session INTERNAL-BGP
    remote-as 50000
timers 30 300
exit-peer-session
```

The following example creates a peer session template named CORE1. This example inherits the configuration of the peer session template named INTERNAL-BGP.

```plaintext
router bgp 45000
template peer-session CORE1
    description CORE-123
    update-source loopback 1
    inherit peer-session INTERNAL-BGP
exit-peer-session
```

The following example configures the 192.168.3.2 neighbor to inherit the CORE1 peer session template. The 192.168.3.2 neighbor will also indirectly inherit the configuration from the peer session template.
named INTERNAL-BGP. The explicit **remote-as** statement is required for the neighbor inherit statement to work. If a peering is not configured, the specified neighbor will not accept the session template.

```
router bgp 45000
neighbor 192.168.3.2 remote-as 50000
neighbor 192.168.3.2 inherit peer-session CORE1
```

**Example Configuring Peer Policy Templates**

The following example creates a peer policy template named GLOBAL in policy-template configuration mode:

```
router bgp 45000
template peer-policy GLOBAL
  weight 1000
  maximum-prefix 5000
  prefix-list NO_SALES in
  exit-peer-policy
```

The following example creates a peer policy template named PRIMARY-IN in policy-template configuration mode:

```
template peer-policy PRIMARY-IN
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  weight 2345
  default-originate
  exit-peer-policy
```

The following example creates a peer policy template named CUSTOMER-A. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
  exit-peer-policy
```

The following example configures the 192.168.2.2 neighbor in address family mode to inherit the peer policy template name CUSTOMER-A. The 192.168.2.2 neighbor will also indirectly inherit the peer policy templates named PRIMARY-IN and GLOBAL.

```
router bgp 45000
neighbor 192.168.2.2 remote-as 50000
address-family ipv4 unicast
neighbor 192.168.2.2 inherit peer-policy CUSTOMER-A
end
```

**Examples Monitoring and Maintaining BGP Dynamic Update Peer-Groups**

No configuration is required to enable the BGP dynamic update of peer groups and the algorithm runs automatically. The following examples show how BGP update group information can be cleared or displayed.

**clear ip bgp update-group Example**

The following example clears the membership of neighbor 10.0.0.1 from an update group:

```
Router#
clear ip bgp update-group 10.0.0.1
```
debug ip bgp groups Example

The following example output from the **debug ip bgp groups** command shows the recalculation of update groups after the **clear ip bgp groups** command was issued:

```
Router# debug ip bgp groups
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.5 flags 0x0 cap 0x0 and updgrp 2 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 2 f10
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.21 flags 0x0 cap 0x0 and updgrp 1 f0
5w4d: BGP-DYN(0): Update-group 1 flags 0x0 cap 0x0 policies same as 10.4.9.21 f0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

show ip bgp replication Example

The following sample output from the **show ip bgp replication** command shows update group replication information for all for neighbors:

```
Router# show ip bgp replication
BGP Total Messages Formatted/Enqueued : 0/0
  Index     Type  Members          Leader   MsgFmt  MsgRepl  Csize  Qsize
  1 internal        1       10.4.9.21        0        0      0      0
  2 internal        2        10.4.9.5        0        0      0      0
```

show ip bgp update-group Example

The following sample output from the **show ip bgp update-group** command shows update group information for all neighbors:

```
Router# show ip bgp update-group
BGP version 4 update-group 1, internal, Address Family: IPv4 Unicast
  BGP Update version : 0, messages 0/0
  Route map for outgoing advertisements is COST1
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 5 seconds
  Has 1 member:
  10.4.9.21
BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 0, messages 0/0
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 5 seconds
  Has 2 members:
  10.4.9.5 10.4.9.8
```

**Where to Go Next**

- If you want to connect to an external service provider, see the "Connecting to a Service Provider Using External BGP" module.
- To configure BGP neighbor session options, proceed to the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some iBGP features, see the "Configuring Internal BGP Features" module.
## Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS IPv6 Command Reference</em></td>
</tr>
<tr>
<td>Overview of Cisco BGP conceptual information with links to all the individual BGP modules</td>
<td>&quot;Cisco BGP Overview&quot; module</td>
</tr>
<tr>
<td>Multiprotocol Label Switching (MPLS) and BGP configuration example using the IPv4 VRF address family</td>
<td>&quot;Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels&quot; module</td>
</tr>
<tr>
<td>Basic MPLS VPN and BGP configuration example</td>
<td>&quot;Configuring MPLS Layer 3 VPNs&quot; module</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT SAFI</td>
<td>MDT SAFI</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-BGP4-MIB</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1772</td>
<td>Application of the Border Gateway Protocol in the Internet</td>
</tr>
<tr>
<td>RFC 1773</td>
<td>Experience with the BGP Protocol</td>
</tr>
</tbody>
</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.

### Feature Information for Configuring a Basic BGP Network

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1774</td>
<td><em>BGP-4 Protocol Analysis</em></td>
</tr>
<tr>
<td>RFC 1930</td>
<td><em>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</em></td>
</tr>
<tr>
<td>RFC 2519</td>
<td><em>A Framework for Inter-Domain Route Aggregation</em></td>
</tr>
<tr>
<td>RFC 2858</td>
<td><em>Multiprotocol Extensions for BGP-4</em></td>
</tr>
<tr>
<td>RFC 2918</td>
<td><em>Route Refresh Capability for BGP-4</em></td>
</tr>
<tr>
<td>RFC 3392</td>
<td><em>Capabilities Advertisement with BGP-4</em></td>
</tr>
<tr>
<td>RFC 4271</td>
<td><em>A Border Gateway Protocol 4 (BGP-4)</em></td>
</tr>
<tr>
<td>RFC 4893</td>
<td><em>BGP Support for Four-octet AS Number Space</em></td>
</tr>
<tr>
<td>RFC 5396</td>
<td><em>Textual Representation of Autonomous system (AS) Numbers</em></td>
</tr>
<tr>
<td>RFC 5398</td>
<td><em>Autonomous System (AS) Number Reservation for Documentation Use</em></td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
### Feature Information for Configuring a Basic BGP Network

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Conditional Route Injection</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Conditional Route Injection feature allows you to inject more specific prefixes into a BGP routing table over less specific prefixes that were selected through normal route aggregation. These more specific prefixes can be used to provide a finer granularity of traffic engineering or administrative control than is possible with aggregated routes.\nThis feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>BGP Configuration Using Peer Templates</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Configuration Using Peer Templates feature introduces a new mechanism that groups distinct neighbor configurations for BGP neighbors that share policies. This type of policy configuration has been traditionally configured with BGP peer groups. However, peer groups have certain limitations because peer group configuration is bound to update grouping and specific session characteristics. Configuration templates provide an alternative to peer group configuration and overcome some of the limitations of peer groups.\nThis feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Dynamic Update Peer Groups</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Dynamic Update Peer Groups feature introduces a new algorithm that dynamically calculates and optimizes update groups of neighbors that share the same outbound policies and can share the same update messages. In previous versions of Cisco IOS XE software, BGP update messages were grouped based on peer-group configurations. This method of grouping updates limited outbound policies and specific-session configurations. The BGP Dynamic Update Peer Group feature separates update group replication from peer group configuration, which improves convergence time and flexibility of neighbor configuration. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP Hybrid CLI</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Hybrid CLI feature simplifies the migration of BGP networks and existing configurations from the NLRI format to the AFI format. This new functionality allows the network operator to configure commands in the AFI format and save these command configurations to existing NLRI formatted configurations. The feature provides the network operator with the capability to take advantage of new features and provides support for migration from the NLRI format to the AFI format. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Neighbor Policy</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Neighbor Policy feature introduces new keywords to two existing commands to display information about local and inherited policies. When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. Inherited policies are policies that the neighbor inherits from a peer-group or a peer-policy template. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were modified by this feature: <code>show ip bgp neighbors</code>, <code>show ip bgp template peer-policy</code>.</td>
</tr>
<tr>
<td>BGP 4 Soft Config</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Support for 4-Byte ASN</td>
<td>Cisco IOS XE Release 2.3 Cisco IOS XE Release 2.4</td>
<td>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295. In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte autonomous system numbers uses asdot as the only configuration format, regular expression match, and output display, with no asplain support. In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the <code>bgp asnotation dot</code> command. The following commands were introduced or modified by this feature: <code>bgp asnotation dot</code>, <code>bgp confederation identifier</code>, <code>bgp confederation peers</code>, <code>all clear ip bgp</code>, <code>commands that configure an autonomous system number</code>, <code>ip as-path access-list</code>, <code>ip extcommunity-list</code>, <code>match source-protocol</code>, <code>neighbor local-as</code>, <code>neighbor remote-as</code>, <code>redistribute (IP)</code>, <code>router bgp</code>, <code>route-target</code>, <code>set as-path</code>, <code>set...</code></td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>extcommunity, set origin, all</td>
<td></td>
<td>extcommunity-list.</td>
</tr>
<tr>
<td>show ip bgp commands that</td>
<td></td>
<td></td>
</tr>
<tr>
<td>display an autonomous system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number, and show ip extcommunity-list.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suppress BGP Advertisement for</td>
<td>Cisco IOS XE Release 2.1</td>
<td>TheSuppressBGPA</td>
</tr>
<tr>
<td>Inactive Routes</td>
<td></td>
<td>Advertisements for Inactive Routes feature allows you to configure the suppression of advertisements for routes that are not installed in the RIB. Configuring this feature allows BGP updates to be more consistent with data used for traffic forwarding.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring Multiprotocol BGP (MP-BGP) Support for CLNS

This module describes configuration tasks to configure multiprotocol BGP (MP-BGP) support for CLNS, which provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks.

- Finding Feature Information, page 123
- Restrictions for Configuring MP-BGP Support for CLNS, page 123
- Information About Configuring MP-BGP Support for CLNS, page 124
- How to Configure MP-BGP Support for CLNS, page 127
- Configuration Examples for MP-BGP Support for CLNS, page 151
- Additional References, page 159
- Feature Information for Configuring MP-BGP Support for CLNS, page 161
- Glossary, page 163

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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.

Restrictions for Configuring MP-BGP Support for CLNS

The configuration of MP-BGP support for CLNS does not support the creation and use of BGP confederations within the CLNS network. We recommend the use of route reflectors to address the issue of a large internal BGP mesh.

BGP extended communities are not supported by the MP-BGP Support for CLNS feature.

The following BGP commands are not supported by the MP-BGP Support for CLNS feature:

- auto-summary
- neighbor advertise-map
• **neighbor distribute-list**
• **neighbor soft-reconfiguration**
• **neighbor unsuppress-map**

### Information About Configuring MP-BGP Support for CLNS

- **Address Family Routing Information**, page 124
- **Design Features of MP-BGP Support for CLNS**, page 124
- **Generic BGP CLNS Network Topology**, page 124
- **DCN Network Topology**, page 126
- **Benefits of MP-BGP Support for CLNS**, page 127

#### Address Family Routing Information

By default, commands entered under the `router bgp` command apply to the IPv4 address family. This will continue to be the case unless you enter the **no bgp default ipv4-unicast** command as the first command under the `router bgp` command. The **no bgp default ipv4-unicast** command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

#### Design Features of MP-BGP Support for CLNS

The configuration of MP-BGP support for CLNS allows BGP to be used as an interdomain routing protocol in networks that use CLNS as the network-layer protocol. This feature was developed to solve a scaling issue with a data communications network (DCN) where large numbers of network elements are managed remotely. For details about the DCN issues and how to implement this feature in a DCN topology, see the **DCN Network Topology**, page 126.

BGP, as an Exterior Gateway Protocol, was designed to handle the volume of routing information generated by the Internet. Network administrators can control the BGP routing information because BGP neighbor relationships (peering) are manually configured and routing updates use incremental broadcasts. Some interior routing protocols such as Intermediate System-to-Intermediate System (IS-IS), in contrast, use a form of automatic neighbor discovery and broadcast updates at regular intervals.

CLNS uses network service access point (NSAP) addresses to identify all its network elements. Using the BGP address-family support, NSAP address prefixes can be transported using BGP. In CLNS, BGP prefixes are inserted into the CLNS Level 2 prefix table. This functionality allows BGP to be used as an interdomain routing protocol between separate CLNS routing domains.

Implementing BGP in routers at the edge of each internal network means that the existing interior protocols need not be changed, minimizing disruption in the network.

#### Generic BGP CLNS Network Topology

The figure below shows a generic BGP CLNS network containing nine routers that are grouped into four different autonomous systems (in BGP terminology) or routing domains (in OSI terminology). To avoid
confusion, we will use the BGP terminology of autonomous systems because each autonomous system is numbered and therefore more easily identified in the diagram and in the configuration discussion.

Figure 12 Components in a Generic BGP CLNS Network

Within each autonomous system, IS-IS is used as the intradomain routing protocol. Between autonomous systems, BGP and its multiprotocol extensions are used as the interdomain routing protocol. Each router is running either a BGP or Level 2 IS-IS routing process. To facilitate this feature, the BGP routers are also running a Level 2 IS-IS process. Although the links are not shown in the figure, each Level 2 IS-IS router is connected to multiple Level 1 IS-IS routers that are, in turn, connected to multiple CLNS networks.

Each autonomous system in this example is configured to demonstrate various BGP features and how these features work with CLNS to provide a scalable interdomain routing solution. In the figure above, the autonomous system AS65101 has a single Level 2 IS-IS router, R1, and is connected to just one other autonomous system, AS65202. Connectivity to the rest of the network is provided by R2, and a default route is generated for R1 to send to R2 all packets with destination NSAP addresses outside of AS65101.

In AS65202 there are two routers, R2 and R3, both with different external BGP (eBGP) neighbors. Routers R2 and R3 are configured to run internal BGP (iBGP) over the internal connection between them.

AS65303 shows how the use of BGP peer groups and route reflection can minimize the need for TCP connections between routers. Fewer connections between routers simplifies the network design and the amount of traffic in the network.

AS65404 shows how to use redistribution to communicate network reachability information to a Level 2 IS-IS router that is not running BGP.

The configuration tasks and examples are based on the generic network design shown in the figure above. Configurations for all the routers in the figure are listed in the Implementing MP-BGP Support for CLNS Example, page 155.
DCN Network Topology

The Multiprotocol BGP (MP-BGP) Support for CLNS feature can benefit a DCN managing a large number of remote SONET rings. SONET is typically used by telecommunications companies to send data over fiber-optic networks.

The figure below shows some components of a DCN network. To be consistent with the BGP terminology, the figure contains labels to indicate three autonomous systems instead of routing domains. The network elements--designated by NE in Figure 2--of a SONET ring are managed by OSI protocols such as File Transfer, Access, and Management (FTAM) and Common Management Information Protocol (CMIP). FTAM and CMIP run over the CLNS network-layer protocol, which means that the routers providing connectivity must run an OSI routing protocol.

**Figure 13** Components in a DCN Network

IS-IS is a link-state protocol used in this example to route CLNS. Each routing node (networking device) is called an intermediate system (IS). The network is divided into areas defined as a collection of routing nodes. Routing within an area is referred to as Level 1 routing. Routing between areas involves Level 2 routing. Routers that link a Level 1 area with a Level 2 area are defined as Level 1-2 routers. A network element that connects to the Level 2 routers that provide a path to the DCN core is represented by a gateway network element--GNE in Figure 2. The network topology here is a point-to-point link between each network element router. In this example, a Level 1 IS-IS router is called an NE router.

Smaller Cisco routers such as the Cisco 2600 series were selected to run as the Level 1-2 routers because shelf space in the central office (CO) of a service provider is very expensive. A Cisco 2600 series router
has limited processing power if it is acting as the Level 1 router for four or five different Level 1 areas. The number of Level 1 areas under this configuration is limited to about 200. The entire Level 2 network is also limited by the speed of the slowest Level 2 router.

To provide connectivity between NE routers, in-band signaling is used. The in-band signaling is carried in the SONET/Synchronous Digital Hierarchy (SDH) frame on the data communications channel (DCC). The DCC is a 192-KB channel, which is a very limited amount of bandwidth for the management traffic. Due to the limited signaling bandwidth between network elements and the limited amount of processing power and memory in the NE routers running IS-IS, each area is restricted to a maximum number of 30 to 40 routers. On average, each SONET ring consists of 10 to 15 network elements.

With a maximum of 200 areas containing 10 to 15 network elements per area, the total number of network element routers in a single autonomous system must be fewer than 3000. Service providers are looking to implement over 10,000 network elements as their networks grow, but the potential number of network elements in an area is limited. The current solution is to break down the DCN into a number of smaller autonomous systems and connect them using static routes or ISO Interior Gateway Routing Protocol (IGRP). ISO IGRP is a proprietary protocol that can limit future equipment implementation options. Static routing does not scale because the growth in the network can exceed the ability of a network administrator to maintain the static routes. BGP has been shown to scale to over 100,000 routes.

To implement the Multiprotocol BGP (MP-BGP) Support for CLNS feature in this example, configure BGP to run on each router in the DCN core network--AS64800 in Figure 2--to exchange routing information between all the autonomous systems. In the autonomous systems AS64600 and AS64700, only the Level 2 routers will run BGP. BGP uses TCP to communicate with BGP-speaking neighbor routers, which means that both an IP-addressed network and an NSAP-addressed network must be configured to cover all the Level 2 IS-IS routers in the autonomous systems AS64600 and AS64700 and all the routers in the DCN core network.

Assuming that each autonomous system--for example, AS64600 and AS64700 in Figure 2--remains the same size with up to 3000 nodes, we can demonstrate how large DCN networks can be supported with this feature. Each autonomous system advertises one address prefix to the core autonomous system. Each address prefix can have two paths associated with it to provide redundancy because there are two links between each autonomous system and the core autonomous system. BGP has been shown to support 100,000 routes, so the core autonomous system can support many other directly linked autonomous systems because each autonomous system generates only a few routes. We can assume that the core autonomous system can support about 2000 directly linked autonomous systems. With the hub-and-spoke design where each autonomous system is directly linked to the core autonomous system, and not acting as a transit autonomous system, the core autonomous system can generate a default route to each linked autonomous system. Using the default routes, the Level 2 routers in the linked autonomous systems process only a small amount of additional routing information. Multiplying the 2000 linked autonomous systems by the 3000 nodes within each autonomous system could allow up to 6 million network elements.

**Benefits of MP-BGP Support for CLNS**

The Multiprotocol BGP (MP-BGP) Support for CLNS feature adds the ability to interconnect separate OSI routing domains without merging the routing domains, which provides the capability to build very large OSI networks. The benefits of using this feature are not confined to DCN networks, and can be implemented to help scale any network using OSI routing protocols with CLNS.

**How to Configure MP-BGP Support for CLNS**
This section contains the following procedures. It may not be necessary to go through each procedure for your particular network. You must perform the steps in the required procedures, but all other procedures are done as required for your network.

- Configuring and Activating a BGP Neighbor to Support CLNS, page 128
- Configuring an IS-IS Routing Process, page 130
- Configuring Interfaces That Connect to BGP Neighbors, page 131
- Configuring Interfaces Connected to the Local OSI Routing Domain, page 133
- Advertising Networking Prefixes, page 134
- Redistributing Routes from BGP into IS-IS, page 137
- Redistributing Routes from IS-IS into BGP, page 138
- Configuring BGP Peer Groups and Route Reflectors, page 140
- Filtering Inbound Routes Based on NSAP Prefixes, page 142
- Filtering Outbound BGP Updates Based on NSAP Prefixes, page 143
- Originating Default Routes for a Neighboring Routing Domain, page 146
- Verifying MP-BGP Support for CLNS, page 148
- Troubleshooting MP-BGP Support for CLNS, page 150

Configuring and Activating a BGP Neighbor to Support CLNS

To configure and activate a BGP routing process and an associated BGP neighbor (peer) to support CLNS, perform the steps in this procedure.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp as-number
4. no bgp default ipv4-unicast
5. neighbor {ip-address | peer-group-name} remote-as as-number
6. address-family nsap [unicast]
7. neighbor ip-address activate
8. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

**Example:**

Router> enable
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2 configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3 router bgp as-number</strong></td>
<td>Configures a BGP routing process and enters router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# router bgp 65101</td>
</tr>
<tr>
<td><strong>Step 4 no bgp default ipv4-unicast</strong></td>
<td>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# no bgp default ipv4-unicast</td>
</tr>
<tr>
<td>**Step 5 neighbor {ip-address</td>
<td>peer-group-name} remote-as as-number**</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# neighbor 10.1.2.2 remote-as 64202</td>
</tr>
<tr>
<td><strong>Step 6 address-family nsap [unicast]</strong></td>
<td>Specifies the NSAP address family and enters address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# address-family nsap</td>
</tr>
<tr>
<td><strong>Step 7 neighbor ip-address activate</strong></td>
<td>Enables the BGP neighbor to exchange prefixes for the NSAP address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# neighbor 10.1.2.2 activate</td>
</tr>
</tbody>
</table>

**Note** If you have configured a peer group as a BGP neighbor, you do not use this command because peer groups are automatically activated when any peer group parameter is configured.
Configuring an IS-IS Routing Process

When an integrated IS-IS routing process is configured, the first instance of the IS-IS routing process configured is by default a Level 1-2 (intra-area and interarea) router. All subsequent IS-IS routing processes on a network running CLNS are configured as Level 1. All subsequent IS-IS routing processes on a network running IP are configured as Level-1-2. To use the Multiprotocol BGP (MP-BGP) Support for CLNS feature, configure a Level 2 routing process.

To configure an IS-IS routing process and assign it as a Level-2-only process, perform the steps in this procedure.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router isis area-tag
4. net network-entity-title
5. is-type [level-1 | level-1-2 | level-2-only]
6. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Step 3** router isis area-tag | Configures an IS-IS routing process and enters router configuration mode for the specified routing process.  
  - The *area-tag* argument is a meaningful name for a routing process. It must be unique among all IP and CLNS routing processes for a given router. |
| **Example:**           | Router(config)# router isis osi-as-101                                                                                                                                                                   |
| **Step 4** net network-entity-title | Configures a network entity title (NET) for the routing process.  
  - If you are configuring multiarea IS-IS, you must specify a NET for each routing process.                                                                                     |
| **Example:**           | Router(config-router)# net 49.0101.1111.1111.1111.1111.00                                                                                                                                               |
| **Step 5** is-type [level-1 | Configures the router to act as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (interarea) router, or as an interarea router only.  
  - In multiarea IS-IS configurations, the first instance of the IS-IS routing process configured is by default a Level 1-2 (intra-area and interarea) router. All subsequent IS-IS routing processes on a network running CLNS are configured as Level 1. All subsequent IS-IS routing processes on a network running IP are configured as Level 1-2. |
|  | level-1-2 | level-2-only] | Example:  
  | Router(config-router)# is-type level-1 |                                                                                                                                                                                                   |
| **Step 6** end         | Exits router configuration mode and returns to privileged EXEC mode.                                                                                                                                       |
| **Example:**           | Router(config-router)# end                                                                                                                                                                              |

### Configuring Interfaces That Connect to BGP Neighbors

When a router running IS-IS is directly connected to an eBGP neighbor, the interface between the two eBGP neighbors is activated using the `clns enable` command, which allows CLNS packets to be forwarded across the interface. The `clns enable` command activates the End System-to-Intermediate System (ES-IS) protocol to search for neighboring OSI systems.

#### Note
Running IS-IS across the same interface that is connected to an eBGP neighbor can lead to undesirable results if the two OSI routing domains merge into a single domain.

When a neighboring OSI system is found, BGP checks that it is also an eBGP neighbor configured for the NSAP address family. If both the preceding conditions are met, BGP creates a special BGP neighbor route in the CLNS Level 2 prefix routing table. The special BGP neighbor route is automatically redistributed into the Level 2 routing updates so that all other Level 2 IS-IS routers in the local OSI routing domain know how to reach this eBGP neighbor.
To configure interfaces that are being used to connect with eBGP neighbors, perform the steps in this procedure. These interfaces will normally be directly connected to their eBGP neighbor.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number
4. ip address ip-address mask
5. clns enable
6. no shutdown
7. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** configure terminal | Enters global configuration mode.                                       |
| Example:                     |                                                                         |
| Router# configure terminal   |                                                                         |

| **Step 3** interface type number | Specifies the interface type and number and enters interface configuration mode. |
| Example:                       |                                                                         |
| Router(config)# interface serial 2/0/0 |                                                                         |

| **Step 4** ip address ip-address mask | Configures the interface with an IP address.                            |
| Example:                           |                                                                         |
| Router(config-if)# ip address 10.1.2.2 255.255.255.0 |                                                                         |

| **Step 5** clns enable | Specifies that CLNS packets can be forwarded across this interface.     |
| Example:              |                                                                         |
| Router(config-if)#    |                                                                         |
| clns enable           |                                                                         |
### Configuring Interfaces Connected to the Local OSI Routing Domain

To configure interfaces that are connected to the local OSI routing domain, perform the steps in this procedure.

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface type number`
4. `ip address ip-address mask`
5. `clns router isis area-tag`
6. `ip router isis area-tag`
7. `no shutdown`
8. `end`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `enable` | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
<p>| <strong>Example:</strong> | |
| <code>Router&gt; enable</code> | |
| <strong>Step 2</strong> <code>configure terminal</code> | Enters global configuration mode. |
| <strong>Example:</strong> | |
| <code>Router# configure terminal</code> | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> interface type number</td>
<td>Specifies the interface type and number and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface gigabitethernet 0/1/1</td>
</tr>
<tr>
<td><strong>Step 4</strong> ip address ip-address mask</td>
<td>Configures the interface with an IP address.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# ip address 10.2.3.1 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 5</strong> clns router isis area-tag</td>
<td>Specifies that the interface is actively routing IS-IS when the network protocol is ISO CLNS and identifies the area associated with this routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# clns router isis osi-as-202</td>
</tr>
<tr>
<td><strong>Step 6</strong> ip router isis area-tag</td>
<td>Specifies that the interface is actively routing IS-IS when the network protocol is IP and identifies the area associated with this routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# ip router isis osi-as-202</td>
</tr>
<tr>
<td><strong>Step 7</strong> no shutdown</td>
<td>Turns on the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# no shutdown</td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# end</td>
</tr>
</tbody>
</table>

**Advertising Networking Prefixes**

Advertising NSAP address prefix forces the prefixes to be added to the BGP routing table. To configure advertisement of networking prefixes, perform the steps in this procedure.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp as-number
4. no bgp default ipv4-unicast
5. neighbor {ip-address | peer-group-name} remote-as as-number
6. address-family nsap [unicast]
7. network nsap-prefix [route-map map-tag]
8. neighbor ip-address activate
9. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp as-number</td>
<td>Configures a BGP routing process and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 65101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 5
**neighbor** *(ip-address | peer-group-name)* **remote-as** *as-number*

**Example:**

```bash
Router(config-router)# neighbor 10.1.2.2 remote-as 64202
```

**Purpose:** Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.

#### Step 6
**address-family nsap [unicast]**

**Example:**

```bash
Router(config-router)# address-family nsap
```

**Purpose:** Specifies the NSAP address family and enters address family configuration mode.

- The optional `unicast` keyword specifies the NSAP unicast address prefixes. By default, the router is placed in unicast NSAP address family configuration mode if the `unicast` keyword is not specified with the `address-family nsap` command.

#### Step 7
**network** *nsap-prefix [route-map map-tag]*

**Example:**

```bash
Router(config-router-af)# network 49.0101.1111.1111.1111.1111.00
```

**Purpose:** Advertises a single prefix of the local OSI routing domain and enters it in the BGP routing table.

**Note**
It is possible to advertise a single prefix, in which case this prefix could be the unique NSAP address prefix of the local OSI routing domain. Alternatively, multiple longer prefixes, each covering a small portion of the OSI routing domain, can be used to selectively advertise different areas.

- The advertising of NSAP address prefixes can be controlled by using the optional `route-map` keyword. If no route map is specified, all NSAP address prefixes are redistributed.

#### Step 8
**neighbor** *ip-address activate*

**Example:**

```bash
Router(config-router-af) neighbor 10.1.2.2 activate
```

**Purpose:** Specifies that NSAP routing information will be sent to the specified BGP neighbor.

**Note**
See the description of the `neighbor` command in the documents listed in the "Additional References" for more details on the use of this command.

#### Step 9
**end**

**Example:**

```bash
Router(config-router-af)# end
```

**Purpose:** Exits address family configuration mode and returns to privileged EXEC mode.
Redistributing Routes from BGP into IS-IS

Route redistribution must be approached with caution. We do not recommend injecting the full set of BGP routes into IS-IS because excessive routing traffic will be added to IS-IS. Route maps can be used to control which dynamic routes are redistributed.

To configure route redistribution from BGP into IS-IS, perform the steps in this procedure.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router isis area-tag
4. net network-entity-title
5. redistribute protocol as-number [route-type] [route-map map-tag]
6. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 enable</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2 configure terminal**       | Enters global configuration mode. |
| Example:                            |         |
| Router# configure terminal          |         |

| **Step 3 router isis area-tag**     | Configures an IS-IS routing process and enters router configuration mode for the specified routing process. |
| Example:                            | You cannot redistribute BGP routes into a Level 1-only IS-IS routing process. |
| Router(config)# router isis osi-as-404 |         |

| **Step 4 net network-entity-title** | Configures a NET for the routing process. |
| Example:                            | If you are configuring multiarea IS-IS, you must specify a NET for each routing process. |
| Router(config-router)# net 49.0404.7777.7777.7777.7777.00 |         |
Redistributing Routes from IS-IS into BGP

Route redistribution must be approached with caution because redistributed route information is stored in the routing tables. Large routing tables may make the routing process slower. Route maps can be used to control which dynamic routes are redistributed.

To configure route redistribution from IS-IS into BGP, perform the steps in this procedure.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp as-number
4. no bgp default ipv4-unicast
5. address-family nsap [unicast]
6. redistribute protocol [process-id] [route-type] [route-map map-tag]
7. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

```
Command or Action                  Purpose
Step 5 redistribute protocol as-number [route-type] [route-map map-tag] | Redistributes NSAP prefix routes from BGP into the CLNS Level 2 routing table associated with the IS-IS routing process when the protocol argument is set to bgp and the route-type argument is set to clns. |
| Example:  |         |
| Router(config-router)# redistribute bgp 65404 clns | \- The as-number argument is defined as the autonomous system number of the BGP routing process to be redistributed into CLNS.  
\- The redistribution of routes can be controlled by using the optional route-map keyword. If no route map is specified, all BGP routes are redistributed. |
| Step 6 end | Exits router configuration mode and returns to privileged EXEC mode. |
| Example:  |         |
| Router(config-router)# end |         |
```
## Configuring Multiprotocol BGP (MP-BGP) Support for CLNS

How to Configure MP-BGP Support for CLNS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp</code> <code>as-number</code></td>
<td>Configures a BGP routing process and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# router bgp 65202</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>no bgp default ipv4-unicast</code></td>
<td>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router)# no bgp default ipv4-unicast</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>address-family nsap [unicast]</code></td>
<td>Specifies the NSAP address family and enters address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router)# address-family nsap</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>redistribute</code> <code>protocol</code> <code>[process-id]</code> <code>[route-type]</code> <code>[route-map</code> <code>map-tag]</code></td>
<td>Redistributes routes from the CLNS Level 2 routing table associated with the IS-IS routing process into BGP as NSAP prefixes when the <code>protocol</code> argument is set to <code>isis</code> and the <code>route-type</code> argument is set to <code>clns</code>.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router-af)# redistribute isis osi-as-202 clns route-map internal-routes-only</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>end</code></td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router-af)# end</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring BGP Peer Groups and Route Reflectors

BGP peer groups reduce the number of configuration commands by applying a BGP neighbor command to multiple neighbors. Using a BGP peer group with a local router configured as a BGP route reflector allows BGP routing information received from one member of the group to be replicated to all other group members. Without a peer group, each route reflector client must be specified by IP address.

To create a BGP peer group and use the group as a BGP route reflector client, perform the steps in this procedure. This is an optional task and is used with internal BGP neighbors. In this task, some of the BGP syntax is shown with the peer-group-name argument only and only one neighbor is configured as a member of the peer group. Repeat Step 9 to configure other BGP neighbors as members of the peer group.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp as-number
4. no bgp default ipv4-unicast
5. neighbor peer-group-name peer-group
6. neighbor peer-group-name remote-as as-number
7. address-family nsap [unicast]
8. neighbor peer-group-name route-reflector-client
9. neighbor ip-address peer-group peer-group
10. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 router bgp as-number</td>
<td>Configures a BGP routing process and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 65303</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 4
**no bgp default ipv4-unicast**

**Example:**

```plaintext
Router(config-router)# no bgp default ipv4-unicast
```

**Purpose:** Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

#### Step 5
**neighbor peer-group-name peer-group**

**Example:**

```plaintext
Router(config-router)# neighbor ibgp-peers peer-group
```

**Purpose:** Creates a BGP peer group.

#### Step 6
**neighbor peer-group-name remote-as as-number**

**Example:**

```plaintext
Router(config-router)# neighbor ibgp-peers remote-as 65303
```

**Purpose:** Adds the peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.

#### Step 7
**address-family nsap [unicast]**

**Example:**

```plaintext
Router(config-router)#
address-family nsap
```

**Purpose:** Specifies the NSAP address family and enters address family configuration mode.

#### Step 8
**neighbor peer-group-name route-reflector-client**

**Example:**

```plaintext
Router(config-router-af)#
neighbor ibgp-peers route-reflector-client
```

**Purpose:** Configures the router as a BGP route reflector and configures the specified peer group as its client.

#### Step 9
**neighbor ip-address peer-group peer-group**

**Example:**

```plaintext
Router(config-router-af)#
neighbor 10.4.5.4 peer-group ibgp-peers
```

**Purpose:** Assigns a BGP neighbor to a BGP peer group.
Filtering Inbound Routes Based on NSAP Prefixes

Perform this task to filter inbound BGP routes based on NSAP prefixes. The `neighbor prefix-list in` command is configured in address family configuration mode to filter inbound routes.

You must specify either a CLNS filter set or a CLNS filter expression before configuring the `neighbor` command. See descriptions for the `clns filter-expr` and `clns filter-set` commands for more information.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp as-number`
4. `no bgp default ipv4-unicast`
5. `address-family nsap [unicast]`
6. `neighbor [ip-address|peer-group-name] prefix-list {clns-filter-expr-name|clns-filter-set-name} in`
7. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1  <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td>Step 2  <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp as-number</td>
<td>Configures a BGP routing process and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 65200</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family nsap [unicast]</td>
<td>Specifies the address family and enters address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family nsap</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor {ip-address</td>
<td>peer-group-name}prefix-list {clns-filter-exp-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.23.4.1 prefix-list abc in</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Filtering Outbound BGP Updates Based on NSAP Prefixes**

Perform this task to filter outbound BGP updates based on NSAP prefixes, use the **neighbor prefix-list out** command in address family configuration mode. This task is configured at Router 7 in the figure above (in the "Generic BGP CLNS Network Topology" section). In this task, a CLNS filter is created with two entries to deny NSAP prefixes starting with 49.0404 and to permit all other NSAP prefixes starting with 49. A BGP peer group is created and the filter is applied to outbound BGP updates for the neighbor that is a member of the peer group.
SUMMARY STEPS

1. enable
2. configure terminal
3. clns filter-set name [deny] template
4. clns filter-set name [permit] template
5. router bgp as-number
6. no bgp default ipv4-unicast
7. neighbor peer-group-name peer-group
8. neighbor {ip-address | peer-group-name} remote-as as-number
9. address-family nsap [unicast]
10. neighbor {ip-address | peer-group-name} prefix-list {clns-filter-expr-name | clns-filter-set-name} out
11. neighbor ip-address peer-group peer-group
12. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> clns filter-set name [deny] template</td>
<td>Defines a NSAP prefix match for a deny condition for use in CLNS filter expressions.</td>
</tr>
<tr>
<td>Example:</td>
<td>• In this example, a deny action is returned if an address starts with 49.0404.</td>
</tr>
<tr>
<td>Router(config)# clns filter-set routes0404 deny 49.0404...</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 4</strong> clns filter-set name [permit] template</td>
<td>Defines a NSAP prefix match for a permit condition for use in CLNS filter expressions.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# clns filter-set routes0404 permit 49...</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Although the permit example in this step allows all NSAP addresses starting with 49, the match condition in Step 3 is processed first so the NSAP addresses starting with 49.0404 are still denied.</td>
</tr>
<tr>
<td><strong>Step 5</strong> router bgp as-number</td>
<td>Configures a BGP routing process and enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 65404</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> no bgp default ipv4-unicast</td>
<td>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor peer-group-name peer-group</td>
<td>Creates a BGP peer group.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor ebgp-peers peer-group</td>
<td>In this example, the BGP peer group named ebgp-peers is created.</td>
</tr>
<tr>
<td><strong>Step 8</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as as-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor ebgp-peers remote-as 65303</td>
<td>In this example, the peer group named ebgp-peers is added to the BGP neighbor table.</td>
</tr>
<tr>
<td><strong>Step 9</strong> address-family nsap [unicast]</td>
<td>Specifies the NSAP address family and enters address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family nsap</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 10** neighbor {ip-address | peer-group-name} prefix-list {clns-filter-expr-name | clns-filter-set-name} out | Specifies a CLNS filter set or CLNS filter expression to be used to filter outbound BGP updates.  
- The clns-filter-expr-name argument is defined with the clns filter-expr configuration command.  
- The clns-filter-set-name argument is defined with the clns filter-set configuration command.  
- In this example, the filter set named routes0404 was created in Step3 and Step 4. |
| **Example:** | |
| Router(config-router-af)# neighbor ebgp-peers prefix-list routes0404 out | |
| **Step 11** neighbor ip-address peer-group peer-group | Assigns a BGP neighbor to a BGP peer group. |
| **Example:** | |
| Router(config-router-af)# neighbor 10.6.7.8 peer-group ebgp-peers | |
| **Step 12** end | Exits address family configuration mode and returns to privileged EXEC mode. |
| **Example:** | |
| Router(config-router-af)# end | |

**Originating Default Routes for a Neighboring Routing Domain**

To create a default CLNS route that points to the local router on behalf of a neighboring OSI routing domain, perform the steps in this procedure. This is an optional task and is normally used only with external BGP neighbors.

**SUMMARY STEPS**

1. enable  
2. configure terminal  
3. router bgp as-number  
4. no bgp default ipv4-unicast  
5. address-family nsap [unicast]  
6. neighbor {ip-address | peer-group-name} default-originate [route-map map-tag]  
7. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:** Router> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal | |
| **Step 3** router bgp as-number | Configures a BGP routing process and enters router configuration mode for the specified routing process. |
| **Example:** Router(config)# router bgp 64803 | |
| **Step 4** no bgp default ipv4-unicast | Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers. |
| **Example:** Router(config-router)# no bgp default ipv4-unicast | |
| **Step 5** address-family nsap [unicast] | Specifies the NSAP address family and enters address family configuration mode. |
| **Example:** Router(config-router)# address-family nsap | |
| **Step 6** neighbor {ip-address | peer-group-name} default-originate [route-map map-tag] | Generates a default CLNS route that points to the local router and that will be advertised to the neighboring OSI routing domain. |
| **Example:** Router(config-router-af)# neighbor 172.16.2.3 default-originate | |
Verifying MP-BGP Support for CLNS

To verify the configuration, use the `show running-config` EXEC command. Sample output is located in the Implementing MP-BGP Support for CLNS Example, page 155. To verify that the Multiprotocol BGP (MP-BGP) Support for CLNS feature is working, perform the following steps.

**SUMMARY STEPS**

1. `show clns neighbors`
2. `show clns route`
3. `show bgp nsap unicast summary`
4. `show bgp nsap unicast`

**DETAILED STEPS**

**Step 1**

`show clns neighbors`

Use this command to confirm that the local router has formed all the necessary IS-IS adjacencies with other Level 2 IS-IS routers in the local OSI routing domain. If the local router has any directly connected external BGP peers, the output from this command will show that the external neighbors have been discovered, in the form of ES-IS adjacencies.

In the following example, the output is displayed for router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section). R2 has three CLNS neighbors. R1 and R4 are ES-IS neighbors because these nodes are in different autonomous systems from R2. R3 is an IS-IS neighbor because it is in the same autonomous system as R2. Note that the system ID is replaced by CLNS hostnames (r1, r3, and r4) that are defined at the start of each configuration file. Specifying the CLNS hostname means that you need not remember which system ID corresponds to which hostname.

**Example:**

```
Router# show clns neighbors
Tag osi-as-202:
System Id Interface SNPA            State Holdtime Type Protocol
r1    Se2/0   *HDL*              Up   274      IS   ES-IS
r3    Et0/1   0002.16de.8481     Up   9        L2   IS-IS
r4    Se2/2   *HDL*              Up   275      IS   ES-IS
```

**Step 2**

`show clns route`

Use this command to confirm that the local router has calculated routes to other areas in the local OSI routing domain. In the following example of output from router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section).
Topologies section), the routing table entry—i 49.0202.3333 [110/10] via R3—shows that router R2 knows about other local IS-IS areas within the local OSI routing domain.

Example:

```
Router# show clns route
Codes: C - connected, S - static, d - DecnetIV
      I - ISO-IGRP, i - IS-IS, e - ES-IS
      B - BGP,   b - eBGP-neighbor
C  49.0202.2222 [2/0], Local IS-IS Area
C  49.0202.2222.2222.2222.2222.00 [1/0], Local IS-IS NET
  via r1, Serial2/0
i  49.0202.3333 [110/10]
   via r3, GigabitEthernet0/1/1
b  49.0101.1111.1111.1111.1111.00 [15/10]
   via r1, Serial2/0
b  49.0101 [20/1]
   via r1, Serial2/0
B  49.0303 [20/1]
   via r4, Serial2/0
B  49.0404 [200/1]
   via r9
i  49.0404.9999.9999.9999.9999.00 [110/10]
   via r3, GigabitEthernet0/1/1
```

Step 3  
**show bgp nsap unicast summary**

Use this command to verify that the TCP connection to a particular neighbor is active. In the following example output, search the appropriate row based on the IP address of the neighbor. If the State/PfxRcd column entry is a number, including zero, the TCP connection for that neighbor is active.

Example:

```
Router# show bgp nsap unicast summary
BGP router identifier 10.1.57.11, local AS number 65202
BGP table version is 6, main routing table version 6
5 network entries and 8 paths using 1141 bytes of memory
6 BGP path attribute entries using 360 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 5/0 prefixes, 8/0 paths, scan interval 60 secs
Neighbor  V  AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.1.2.1  4 65101  34   34    6   0 00:29:11  1
10.2.3.3  4 65101  35   36    6   0 00:29:16  3
```

Step 4  
**show bgp nsap unicast**

Enter the `show bgp nsap unicast` command to display all the NSAP prefix routes that the local router has discovered. In the following example output of router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), a single valid route to prefix 49.0101 is shown. Two valid routes—marked by a *—are shown for the prefix 49.0404. The second route is marked with a *>i sequence, representing the best route to this prefix.

Example:

```
Router# show bgp nsap unicast
BGP table version is 3, local router ID is 192.168.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
                Network Next Hop Metric LocPrf Weight Path
*> 49.0101  49.0101.1111.1111.1111.1111.00 0 65101 i
```
Troubleshooting MP-BGP Support for CLNS

The `debug bgp nsap unicast` commands enable diagnostic output concerning various events relating to the operation of the CLNS packets in the BGP routing protocol to be displayed on a console. These commands are intended only for troubleshooting purposes because the volume of output generated by the software when they are used can result in severe performance degradation on the router. See the Cisco IOS Debug Command Reference for more information about using these `debug` commands.

To troubleshoot problems with the configuration of MP-BGP support for CLNS and to minimize the impact of the `debug` commands used in this procedure, perform the following steps.

**SUMMARY STEPS**

1. Attach a console directly to a router running the Cisco IOS XE software release that includes the Multiprotocol BGP (MP-BGP) Support for CLNS feature.
2. `no logging console`
3. Use Telnet to access a router port.
4. `enable`
5. `terminal monitor`
6. `debug bgp nsap unicast [neighbor-address | dampening | keepalives | updates]`
7. `no terminal monitor`
8. `no debug bgp nsap unicast [neighbor-address | dampening | keepalives | updates]`
9. `logging console`

**DETAILED STEPS**

**Step 1**

Attach a console directly to a router running the Cisco IOS XE software release that includes the Multiprotocol BGP (MP-BGP) Support for CLNS feature.

**Note** This procedure will minimize the load on the router created by the `debug bgp nsap unicast` commands because the console port will no longer be generating character-by-character processor interrupts. If you cannot connect to a console directly, you can run this procedure via a terminal server. If you must break the Telnet connection, however, you may not be able to reconnect because the router may be unable to respond due to the processor load of generating the `debug bgp nsap unicast` output.

**Step 2**

`no logging console`
This command disables all logging to the console terminal.

**Step 3**
Use Telnet to access a router port.

**Step 4**
*enable*
Enter this command to access privileged EXEC mode.

**Step 5**
*terminal monitor*
This command enables logging on the virtual terminal.

**Step 6**
*debug bgp nsap unicast [neighbor-address | dampening | keepalives | updates]*
Enter only specific *debug bgp nsap unicast* commands to isolate the output to a certain subcomponent and minimize the load on the processor. Use appropriate arguments and keywords to generate more detailed debug information on specified subcomponents.

**Step 7**
*no terminal monitor*
This command disables logging on the virtual terminal.

**Step 8**
*no debug bgp nsap unicast [neighbor-address | dampening | keepalives | updates]*
Enter the specific *no debug bgp nsap unicast* command when you are finished.

**Step 9**
*logging console*
This command reenables logging to the console.

### Configuration Examples for MP-BGP Support for CLNS

This section provides configuration examples to match the identified configuration tasks in the previous section. To provide an overview of all the router configurations in the figure above (in the "Generic BGP CLNS Network Topology" section), more detailed configurations for each router are added at the end of this section.

- Configuring and Activating a BGP Neighbor to Support CLNS Example, page 151
- Configuring an IS-IS Routing Process Example, page 152
- Configuring Interfaces Example, page 152
- Advertising Networking Prefixes Example, page 152
- Redistributing Routes from BGP into IS-IS Example, page 152
- Redistributing Routes from IS-IS into BGP Example, page 153
- Configuring BGP Peer Groups and Route Reflectors Example, page 153
- Filtering Inbound Routes Based on NSAP Prefixes Example, page 154
- Filtering Outbound BGP Updates Based on NSAP Prefixes Example, page 154
- Originating a Default Route and Outbound Route Filtering Example, page 154
- Implementing MP-BGP Support for CLNS Example, page 155

### Configuring and Activating a BGP Neighbor to Support CLNS Example

In the following example, the router R1, shown in the figure below, in the autonomous system AS65101 is configured to run BGP and activated to support CLNS. Router R1 is the only Level 2 IS-IS router in autonomous system AS65101, and it has only one connection to another autonomous system via router R2 in AS65202. The *no bgp default ipv4-unicast* command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers. After the NSAP address family configuration mode is enabled with the *address-family nsap* command, the
router is configured to advertise the NSAP prefix of 49.0101 to its BGP neighbors and to send NSAP routing information to the BGP neighbor at 10.1.2.2.

```
router bgp 65101
no bgp default ipv4-unicast
address-family nsap
    network 49.0101...
    neighbor 10.1.2.2 activate
    exit-address-family
```

**Configuring an IS-IS Routing Process Example**

In the following example, the router R1, shown in the figure below, is configured to run an IS-IS process:

```
router isis osi-as-101
    net 49.0101.1111.1111.1111.1111.00
```

The default IS-IS routing process level is used.

**Configuring Interfaces Example**

In the following example, two of the interfaces of the router R2, shown in the figure below, in the autonomous system AS65202 are configured to run CLNS. GigabitEthernet interface 0/1/1 is connected to the local OSI routing domain and is configured to run IS-IS when the network protocol is CLNS using the `clns router isis` command. The serial interface 2/0 with the local IP address of 10.1.2.2 is connected with an eBGP neighbor and is configured to run CLNS through the `clns enable` command:

```
interface serial 2/0
    ip address 10.1.2.2 255.255.255.0
    clns enable
    no shutdown

interface gigabitethernet 0/1/1
    ip address 10.2.3.1 255.255.255.0
    clns router isis osi-as-202
    no shutdown
```

**Advertising Networking Prefixes Example**

In the following example, the router R1, shown in the figure below, is configured to advertise the NSAP prefix of 49.0101 to other routers. The NSAP prefix unique to autonomous system AS65101 is advertised to allow the other autonomous systems to discover the existence of autonomous system AS65101 in the network.

```
router bgp 65101
    no bgp default ipv4-unicast
    neighbor 10.1.2.2 remote-as 64202
    address-family nsap
    network 49.0101...
    neighbor 10.1.2.2 activate
```

**Redistributing Routes from BGP into IS-IS Example**

In the following example, the routers R7 and R9, shown in the figure below, in the autonomous system AS65404 are configured to redistribute BGP routes into the IS-IS routing process called osi-as-404. Redistributing the BGP routes allows the Level 2 IS-IS router, R8, to advertise routes to destinations outside the autonomous system AS65404. Without a route map being specified, all BGP routes are redistributed.
Redistributing Routes from IS-IS into BGP Example

In the following example, the router R2, shown in the figure below, in the autonomous system AS65202 is configured to redistribute Level 2 CLNS NSAP routes into BGP. A route map is used to permit only routes from within the local autonomous system to be redistributed into BGP. Without a route map being specified, every NSAP route from the CLNS level 2 prefix table is redistributed. The `no bgp default ipv4-unicast` command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

```
clns filter-set internal-routes permit 49.0202...
  !
  route-map internal-routes-only permit 10
  match clns address internal-routes
  !
  router isis osi-as-202
  net 49.0202.2222.2222.2222.2222.00
  !
  router bgp 65202
  no bgp default ipv4-unicast
  address-family nsap
  redistribute isis osi-as-202 clns route-map internal-routes-only
```

Configuring BGP Peer Groups and Route Reflectors Example

Router R5, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), has only iBGP neighbors and runs IS-IS on both interfaces. To reduce the number of configuration commands, configure R5 as a member of a BGP peer group called ibgp-peers. The peer group is automatically activated under the `address-family nsap` command by configuring the peer group as a route reflector client allowing it to exchange NSAP routing information between group members. The BGP peer group is also configured as a BGP route reflector client to reduce the need for every iBGP router to be linked to each other.

In the following example, the router R5 in the autonomous system AS65303 is configured as a member of a BGP peer group and a BGP route reflector client:

```
router bgp 65303
  no bgp default ipv4-unicast
  neighbor ibgp-peers peer-group
  neighbor ibgp-peers remote-as 65303
  address-family nsap
  neighbor ibgp-peers route-reflector-client
  neighbor 10.4.5.4 peer-group ibgp-peers
  neighbor 10.5.6.6 peer-group ibgp-peers
  exit-address-family
```
Filtering Inbound Routes Based on NSAP Prefixes Example

In the following example, the router R1, shown in the figure below, in the autonomous system AS65101 is configured to filter inbound routes specified by the default-prefix-only prefix list:

```
cl ns filter-set default-prefix-only deny 49...
cl ns filter-set default-prefix-only permit default
!
r ruter r isis osi-as-101
    net 49.0101.1111.1111.1111.1111.00
!
r ruter bgp 65101
    no bgp default ipv4-unicast
    neighbor 10.1.2.2 remote-as 64202
    address-family nsap
    network 49.0101.1111.1111.1111.1111.00
    neighbor 10.1.2.2 activate
    neighbor 10.1.2.2 prefix-list default-prefix-only in
```

Filtering Outbound BGP Updates Based on NSAP Prefixes Example

In the following example, outbound BGP updates are filtered based on NSAP prefixes. This example is configured at Router 7 in the figure below. In this task, a CLNS filter is created with two entries to deny NSAP prefixes starting with 49.0404 and to permit all other NSAP prefixes starting with 49. A BGP peer group is created and the filter is applied to outbound BGP updates for the neighbor that is a member of the peer group.

```
cl ns filter-set routes0404 deny 49.0404...
cl ns filter-set routes0404 permit 49...
!
r ruter bgp 65404
    no bgp default ipv4-unicast
    neighbor ebgp-peers remote-as 65303
    address-family nsap
    neighbor ebgp-peers prefix-list routes0404 out
    neighbor 10.6.7.8 peer-group ebgp-peers
```

Originating a Default Route and Outbound Route Filtering Example

In the following example, the router R2, shown in the figure below, in the autonomous system AS65202 is configured to generate a default route for router R1 in the autonomous system AS65101, and an outbound filter is created to send only the default route NSAP addressing information in the BGP update messages to router R1.

```
cl ns filter-set default-prefix-only deny 49...
cl ns filter-set default-prefix-only permit default
!
r ruter bgp 65202
    no bgp default ipv4-unicast
    neighbor 10.1.2.1 remote-as 64101
    address-family nsap
    network 49.0202...
    neighbor 10.1.2.1 activate
```

Configuration Examples for MP-BGP Support for CLNS
Implementing MP-BGP Support for CLNS Example

The figure below shows a generic BGP CLNS network containing nine routers that are grouped into four different autonomous systems (in BGP terminology) or routing domains (in OSI terminology). This section contains complete configurations for all routers shown in the figure below.

If you need more details about commands used in the following examples, see the configuration tasks earlier in this document and the documents listed in the Additional References, page 159.

- Autonomous System AS65101, page 155
- Autonomous System AS65202, page 156
- Autonomous System AS65303, page 157
- Autonomous System AS65404, page 158

Autonomous System AS65101

Router 1

clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
routing isis osi-as-101
  net 49.0101.1111.1111.1111.1111.1111.00
!
routing bgp 65101
  no bgp default ipv4-unicast
neighbor 10.1.2.2 remote-as 65202
address-family nsa
neighbor 10.1.2.2 activate
neighbor 10.1.2.2 prefix-list default-prefix-only in
network 49.0101...
exit-address-family
! interface serial 2/0
ip address 10.1.2.1 255.255.255.0
clns enable
no shutdown

Autonomous System AS65202

Router 2

clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
match clns address internal-routes
!
router isis osi-as-202
 net 49.0202.2222.2222.2222.2222.00
!
router bgp 65202
 no bgp default ipv4-unicast
neighbor 10.1.2.1 remote-as 65101
neighbor 10.2.3.3 remote-as 65202
neighbor 10.2.4.4 remote-as 65303
address-family nsa
neighbor 10.1.2.1 activate
neighbor 10.2.3.3 activate
neighbor 10.2.4.4 activate
redistribute isis osi-as-202 clns route-map internal-routes-only
neighbor 10.1.2.1 default-originate
neighbor 10.1.2.1 prefix-list default-prefix-only out
exit-address-family
!
interface gigabitethernet 0/1/1
ip address 10.2.3.2 255.255.255.0
clns router isis osi-as-202
no shutdown
!
interface serial 2/0
ip address 10.1.2.2 255.255.255.0
clns enable
no shutdown
!
interface serial 2/2
ip address 10.2.4.2 255.255.255.0
clns enable
no shutdown

Router 3

clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
match clns address internal-routes
!
router isis osi-as-202
 net 49.0202.3333.3333.3333.3333.00
!
router bgp 65202
 no bgp default ipv4-unicast
Autonomous System AS65303

Router 4

```bash
router isis osi-as-303
 net 49.0303.4444.4444.4444.4444.4444.4444.00
!
router bgp 65303
 no bgp default ipv4-unicast
 neighbor 10.2.4.2 remote-as 65202
 neighbor 10.4.5.5 remote-as 65303
 address-family nsap
 no synchronization
 neighbor 10.2.4.2 activate
 neighbor 10.4.5.5 activate
 network 49.0303...
 exit-address-family
!
interface gigabitethernet 0/2/1
 ip address 10.4.5.4 255.255.255.0
 clns router isis osi-as-303
 no shutdown
!
interface serial 2/3
 ip address 10.2.4.4 255.255.255.0
 clns enable
 no shutdown
```

Router 5

```bash
router isis osi-as-303
 net 49.0303.5555.5555.5555.5555.5555.5555.00
!
router bgp 65303
 no bgp default ipv4-unicast
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers remote-as 65303
 address-family nsap
 no synchronization
 neighbor ibgp-peers route-reflector-client
 neighbor 10.4.5.4 peer-group ibgp-peers
 neighbor 10.5.6.6 peer-group ibgp-peers
 exit-address-family
!
interface gigabitethernet 0/2/1
 ip address 10.4.5.5 255.255.255.0
 clns router isis osi-as-303
 no shutdown
!
interface gigabitethernet 0/3/1
```
ip address 10.5.6.5 255.255.255.0
clns router isis osi-as-303
no shutdown

Router 6

router isis osi-as-303
  net 49.0303.6666.6666.6666.6666.00
  router bgp 65303
  no bgp default ipv4-unicast
  neighbor 10.5.6.5 remote-as 65303
  neighbor 10.6.7.7 remote-as 65404
  address-family nsa
  no synchronization
  neighbor 10.5.6.5 activate
  neighbor 10.6.7.7 activate
  network 49.0303...
  interface gigabitethernet 0/3/1
  ip address 10.5.6.6 255.255.255.0
  clns router isis osi-as-303
  no shutdown
  interface serial 2/2
  ip address 10.6.7.6 255.255.255.0
  clns enable
  no shutdown

Autonomous System AS65404

Router 7

clns filter-set external-routes deny 49.0404...
clns filter-set external-routes permit 49...
  route-map noexport permit 10
  match clns address external-routes
  set community noexport
  router isis osi-as-404
  net 49.0404.7777.7777.7777.7777.00
  redistribute bgp 404 clns
  router bgp 65404
  no bgp default ipv4-unicast
  neighbor 10.6.7.6 remote-as 65303
  neighbor 10.8.9.9 remote-as 65404
  address-family nsa
  neighbor 10.6.7.6 activate
  neighbor 10.8.9.9 activate
  neighbor 10.8.9.9 send-community
  neighbor 10.8.9.9 route-map noexport out
  network 49.0404...
  interface gigabitethernet 1/0/1
  ip address 10.7.8.7 255.255.255.0
  clns router isis osi-as-404
  ip router isis osi-as-404
  no shutdown
  interface serial 2/3
  ip address 10.6.7.7 255.255.255.0
  clns enable
  no shutdown
Router 8

router isis osi-as-404
net 49.0404.8888.8888.8888.8888.00
!
interface gigabitethernet 1/0/1
  ip address 10.7.8.8 255.255.255.0
  clns router isis osi-as-404
  ip router isis osi-as-404
  no shutdown
!
interface gigabitethernet 1/1/1
  ip address 10.8.9.8 255.255.255.0
  clns router isis osi-as-404
  ip router isis osi-as-404
  no shutdown

Router 9

clns filter-set external-routes deny 49.0404...
clns filter-set external-routes permit 49...
!
route-map noexport permit 10
  match clns address external-routes
  set community noexport
!
router isis osi-as-404
net 49.0404.9999.9999.9999.9999.00
redistribute bgp 404 clns
!
router bgp 65404
  no bgp default ipv4-unicast
  neighbor 10.3.9.3 remote-as 65202
  neighbor 10.7.8.7 remote-as 65404
  address-family nsap
    network 49.0404...
    neighbor 10.3.9.3 activate
    neighbor 10.7.8.7 activate
    neighbor 10.7.8.7 send-community
    neighbor 10.7.8.7 route-map noexport out
!
interface serial 2/3
  ip address 10.3.9.9 255.255.255.0
  clns enable
  no shutdown
!
interface gigabitethernet 1/1/1
  ip address 10.8.9.9 255.255.255.0
  clns router isis osi-as-404
  ip router isis osi-as-404
  no shutdown

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>CLNS commands</td>
<td>Cisco IOS ISO CLNS Command Reference</td>
</tr>
</tbody>
</table>
### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO/IEC 9542</td>
<td>End System to Intermediate System Protocol (ESIS). End system to Intermediate system routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473).</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1700</td>
<td>Assigned Numbers</td>
</tr>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 1997</td>
<td>BGP Communities Attribute</td>
</tr>
<tr>
<td>RFC 2042</td>
<td>Registering New BGP Attribute Types</td>
</tr>
<tr>
<td>RFC 2439</td>
<td>BGP Route Flap Dampening</td>
</tr>
<tr>
<td>RFC 2842</td>
<td>Capabilities Advertisement with BGP-4</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index">http://www.cisco.com/cisco/web/support/index</a></td>
</tr>
</tbody>
</table>

Feature Information for Configuring MP-BGP Support for CLNS

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 12  Feature Information for MP-BGP Support for CLNS

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Multiprotocol BGP (MP-BGP) Support for CLNS      | Cisco IOS XE Release 2.6 | The Multiprotocol BGP (MP-BGP) Support for CLNS feature provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks. The following commands were introduced or modified by this feature:  
  • address-family nsap  
  • clear bgp nsap  
  • clear bgp nsap dampening  
  • clear bgp nsap external  
  • clear bgp nsap flap-statistics  
  • clear bgp nsap peer-group  
  • debug bgp nsap  
  • debug bgp nsap dampening  
  • debug bgp nsap updates  
  • neighbor prefix-list  
  • network (BGP and multiprotocol BGP)  
  • redistribute (BGP to ISO ISIS)  
  • redistribute (ISO ISIS to BGP)  
  • show bgp nsap  
  • show bgp nsap community  
  • show bgp nsap community-list  
  • show bgp nsap dampened-paths  
  • show bgp nsap filter-list  
  • show bgp nsap flap-statistics
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap inconsistent-as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap neighbors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap paths</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap quote-regexp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap regexp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• show bgp nsap summary</td>
</tr>
</tbody>
</table>

### Glossary

**address family** -- A group of network protocols that share a common format of network address. Address families are defined by RFC 1700.

**AS** -- autonomous system. An IP term to describe a routing domain that has its own independent routing policy and is administered by a single authority. Equivalent to the OSI term "routing domain."

**BGP** -- Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems.

**CLNS** -- Connectionless Network Service. An OSI network-layer protocol.

**CMIP** -- Common Management Information Protocol. In OSI, a network management protocol created and standardized by ISO for the monitoring and control of heterogeneous networks.

**DCC** -- data communications channel.

**DCN** -- data communications network.

**ES-IS** -- End System-to-Intermediate System. OSI protocol that defines how end systems (hosts) announce themselves to intermediate systems (routers).

**FTAM** -- File Transfer, Access, and Management. In OSI, an application-layer protocol developed for network file exchange and management between diverse types of computers.

**IGP** -- Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system.

**IGRP** -- Interior Gateway Routing Protocol. A proprietary Cisco protocol developed to address the issues associated with routing in large, heterogeneous networks.

**IS** -- intermediate system. Routing node in an OSI network.

**IS-IS** -- Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where routers exchange routing information based on a single metric, to determine network topology.

**ISO** -- International Organization for Standardization. International organization that is responsible for a wide range of standards, including those relevant to networking. ISO developed the Open System Interconnection (OSI) reference model, a popular networking reference model.

**NSAP address** -- network service access point address. The network address format used by OSI networks.

**OSI** -- Open System Interconnection. International standardization program created by ISO and ITU-T to develop standards for data networking that facilitate multivendor equipment interoperability.
routing domain -- The OSI term that is equivalent to autonomous system for BGP.

SDH -- Synchronous Digital Hierarchy. Standard that defines a set of rate and format standards that are sent using optical signals over fiber.

SONET -- Synchronous Optical Network. High-speed synchronous network specification designed to run on optical fiber.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Connecting to a Service Provider Using External BGP

This module describes configuration tasks that will enable your Border Gateway Protocol (BGP) network to access peer devices in external networks such as those from Internet service providers (ISPs). BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. External BGP (eBGP) peering sessions are configured to allow peers from different autonomous systems to exchange routing updates. Tasks to help manage the traffic that is flowing inbound and outbound are described, as are tasks to configure BGP policies to filter the traffic. Multihoming techniques that provide redundancy for connections to a service provider are also described.

- Finding Feature Information, page 165
- Prerequisites for Connecting to a Service Provider Using External BGP, page 165
- Restrictions for Connecting to a Service Provider Using External BGP, page 166
- Information About Connecting to a Service Provider Using External BGP, page 166
- How to Connect to a Service Provider Using External BGP, page 177
- Configuration Examples for Connecting to a Service Provider Using External BGP, page 233
- Where to Go Next, page 244
- Additional References, page 245
- Feature Information for Connecting to a Service Provider Using External BGP, page 246

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Connecting to a Service Provider Using External BGP

- Before connecting to a service provider you need to understand how to configure the basic BGP process and peers. See the "Cisco BGP Overview" and "Configuring a Basic BGP Network" modules for more details.
The tasks and concepts in this chapter will help you configure BGP features that would be useful if you are connecting your network to a service provider. For each connection to the Internet, you must have an assigned autonomous system number from the Internet Assigned Numbers Authority (IANA).

Restrictions for Connecting to a Service Provider Using External BGP

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.

Information About Connecting to a Service Provider Using External BGP

- External BGP Peering, page 166
- BGP Autonomous System Number Formats, page 167
- BGP Attributes, page 170
- Multihoming, page 171
- MED Attribute, page 172
- Transit Versus Nontransit Traffic, page 172
- BGP Policy Configuration, page 172
- BGP Communities, page 173
- Extended Communities, page 174
- Extended Community Lists, page 175
- Administrative Distance, page 175
- BGP Route Map Policy Lists, page 175
- BGP Route Map with a Continue Clause, page 176

External BGP Peering

BGP is an interdomain routing protocol designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol and it uses TCP (port 179) as the transport protocol. The destination TCP port is assigned 179, and the local port is assigned a random port number. Cisco IOS software supports BGP version 4, which has been used by ISPs to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use.

External BGP peering sessions are configured to allow BGP peers from different autonomous systems to exchange routing updates. By design, a BGP routing process expects eBGP peers to be directly connected, for example, over a WAN connection. However, there are many real-world scenarios where this rule would prevent routing from occurring. Peering sessions for multihop neighbors are configured with the `neighbor ebgp-multihop` command. The figure below shows simple eBGP peering between three routers. Router B peers with Router A and Router E. In the figure below, the `neighbor ebgp-multihop` command could be used to establish peering between Router A and Router E although this is a very simple network design. BGP forwards information about the next hop in the network using the NEXT_HOP attribute, which is set...
to the IP address of the interface that advertises a route in an eBGP peering session by default. The source interface can be a physical interface or a loopback interface.

**Figure 15 BGP Peers in Different Autonomous Systems**

Loopback interfaces are preferred for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. Interfaces on networking devices can fail, and they can also be taken out of service for maintenance. When an interface is administratively brought up or down, due to failure or maintenance, it is referred to as a flap. Loopback interfaces provide a stable source interface to ensure that the IP address assigned to the interface is always reachable as long as the IP routing protocols continue to advertise the subnet assigned to the loopback interface. Loopback interfaces allow you to conserve address space by configuring a single address with /32 bit mask. Before a loopback interface is configured for an eBGP peering session, you must configure the `neighbor update-source` command and specify the loopback interface. With this configuration, the loopback interface becomes the source interface and its IP address is advertised as the next hop for routes that are advertised through this loopback. If loopback interfaces are used to connect single-hop eBGP peers, you must configure the `neighbor disable-connected-check` command before you can establish the eBGP peering session.

Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet. Traffic will also be flowing into, and possibly through, your network. BGP contains various techniques to influence how the traffic flows into and out of your network, and to create BGP policies that filter the traffic, inbound and outbound. To influence the traffic flow, BGP uses certain BGP attributes that can be included in update messages or used by the BGP routing algorithm. BGP policies to filter traffic also use some of the BGP attributes with route maps, access lists including AS-path access lists, filter lists, policy lists, and distribute lists. Managing your external connections may involve multihoming techniques where there is more than one connection to an ISP or connections to more than one ISP for backup or performance purposes. Tagging BGP routes with different community attributes across autonomous system or physical boundaries can prevent the need to configure long lists of individual permit or deny statements.

**BGP Autonomous System Number Formats**

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were 2-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority...
IANA will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to 4294967295. RFC 5396, Textual Representation of Autonomous System (AS) Numbers, documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- **Asplain**—Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- **Asdot**—Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

### Asdot Only Autonomous System Number Formatting

In Cisco IOS XE Release 2.3, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers the asdot format includes a period, which is a special character in regular expressions. A backslash must be entered before the period; for example, \.14, to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in show command output in Cisco IOS images where only asdot formatting is available.

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

### Asplain as Default Autonomous System Number Formatting

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you want to change the default show command output to display 4-byte autonomous system numbers in the asdot format, use the `bgp asnotation dot` command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte autonomous system numbers in either asplain or asdot format, only one format is used to display show command output and control 4-byte autonomous system number matching for regular expressions, and the default is asplain format. To display 4-byte autonomous system numbers in show command output and to control matching for regular expressions in the asdot format, you must configure the `bgp asnotation dot` command. After enabling the `bgp asnotation dot` command, a hard reset must be initiated for all BGP sessions by entering the `clear ip bgp *` command.
If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The `show` command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

### Table 14  Default Asplain 4-Byte Autonomous System Number Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asplain</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
</tr>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
</tr>
</tbody>
</table>

### Table 15  Asdot 4-Byte Autonomous System Number Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Configuration Format</th>
<th>Show Command Output and Regular Expression Match Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>asplain</td>
<td>2-byte: 1 to 65535 4-byte: 65536 to 4294967295</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
<tr>
<td>asdot</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
<td>2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535</td>
</tr>
</tbody>
</table>

### Reserved and Private Autonomous System Numbers

In Cisco IOS XE Release 2.3 and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.
Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous-system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: http://www.iana.org/.

BGP Attributes

BGP selects a single path, by default, as the best path to a destination host or network. The best-path selection algorithm analyzes path attributes to determine which route is installed as the best path in the BGP routing table. Each path carries various attributes that are used in BGP best-path analysis. Cisco IOS XE software provides the ability to influence BGP path selection by altering these attributes via the command-line interface (CLI). BGP path selection can also be influenced through standard BGP policy configuration.

BGP can include path attribute information in update messages. BGP attributes describe the characteristic of the route, and the software uses these attributes to help make decisions about which routes to advertise. Some of this attribute information can be configured at a BGP-speaking networking device. There are some mandatory attributes that are always included in the update message and some discretionary attributes. The following BGP attributes can be configured:

- AS-path
- Community
- Local_Pref
- Multi_Exit_Discriminator (MED)
- Next_Hop
- Origin

**AS-path**

This attribute contains a list or set of the autonomous system numbers through which routing information has passed. The BGP speaker adds its own autonomous system number to the list when it forwards the update message to external peers.

**Community**

BGP communities are used to group networking devices that share common properties, regardless of network, autonomous system, or any physical boundaries. In large networks applying a common routing policy through prefix lists or access lists requires individual peer statements on each networking device. Using the BGP community attribute BGP neighbors, with common routing policies, can implement inbound or outbound route filters based on the community tag rather than consult large lists of individual permit or deny statements.

**Local_Pref**

Within an autonomous system, the Local_Pref attribute is included in all update messages between BGP peers. If there are several paths to the same destination, the local preference attribute with the highest value indicates the preferred outbound path from the local autonomous system. The highest ranking route is advertised to internal peers. The Local_Pref value is not forwarded to external peers.

**Multi_Exit_Discriminator**

The MED attribute indicates (to an external peer) a preferred path into an autonomous system. If there are multiple entry points into an autonomous system, the MED can be used to influence another autonomous
system to choose one particular entry point. A metric is assigned where a lower MED metric is preferred by the software over a higher MED metric. The MED metric is exchanged between autonomous systems, but after a MED is forwarded into an autonomous system, the MED metric is reset to the default value of 0.

When an update is sent to an internal BGP (iBGP) peer, the MED is passed along without any change, allowing all the peers in the same autonomous system to make a consistent path selection.

By default, a router will compare the MED attribute for paths only from BGP peers that reside in the same autonomous system. The `bgp always-compare-med` command can be configured to allow the router to compare metrics from peers in different autonomous systems.

---

**Note**

The Internet Engineering Task Force (IETF) decision regarding BGP MED assigns a value of infinity to the missing MED, making the route that lacks the MED variable the least preferred. The default behavior of BGP routers that run Cisco IOS XE software is to treat routes without the MED attribute as having a MED of 0, making the route that lacks the MED variable the most preferred. To configure the router to conform to the IETF standard, use the `bgp bestpath med missing-as-worst` router configuration command.

---

**Next_Hop**

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

**Origin**

This attribute indicates how the route was included in a BGP routing table. In Cisco IOS XE software, a route defined using the `bgp network` command is given an origin code of Interior Gateway Protocol (IGP). Routes distributed from an Exterior Gateway Protocol (EGP) are coded with an origin of EGP, and routes redistributed from other protocols are defined as Incomplete. BGP decision policy for origin prefers IGP over EGP, and then EGP over Incomplete.

---

**Multihoming**

Multihoming is defined as connecting an autonomous system with more than one service provider. If you have any reliability issues with one service provider, then you have a backup connection. Performance issues can also be addressed by multihoming because better paths to the destination network can be utilized.

Unless you are a service provider, you must plan your routing configuration carefully to avoid Internet traffic traveling through your autonomous system and consuming all your bandwidth. The figure below shows that autonomous system 45000 is multihomed to autonomous system 40000 and autonomous system 50000. Assuming autonomous system 45000 is not a service provider, then several techniques such as load balancing or some form of routing policy must be configured to allow traffic from autonomous system
45000 to reach either autonomous system 40000 or autonomous system 50000 but not allow much, if any, transit traffic.

**Figure 16 Multihoming Topology**

**MED Attribute**
Configuring the MED attribute is another method that BGP can use to influence the choice of paths into another autonomous system. The MED attribute indicates (to an external peer) a preferred path into an autonomous system. If there are multiple entry points into an autonomous system, the MED can be used to influence another autonomous system to choose one particular entry point. A metric is assigned using route maps where a lower MED metric is preferred by the software over a higher MED metric.

**Transit Versus Nontransit Traffic**
Most of the traffic within an autonomous system contains a source or destination IP address residing within the autonomous system, and this traffic is referred to as nontransit (or local) traffic. Other traffic is defined as transit traffic. As traffic across the Internet increases, controlling transit traffic becomes more important.

A service provider is considered to be a transit autonomous system and must provide connectivity to all other transit providers. In reality, few service providers actually have enough bandwidth to allow all transit traffic, and most service providers have to purchase such connectivity from Tier 1 service providers.

An autonomous system that does not usually allow transit traffic is called a stub autonomous system and will link to the Internet through one service provider.

**BGP Policy Configuration**
BGP policy configuration is used to control prefix processing by the BGP routing process and to filter routes from inbound and outbound advertisements. Prefix processing can be controlled by adjusting BGP timers, altering how BGP handles path attributes, limiting the number of prefixes that the routing process will accept, and configuring BGP prefix dampening. Prefixes in inbound and outbound advertisements are filtered using route maps, filter lists, IP prefix lists, autonomous-system-path access lists, IP policy lists, and distribute lists. The table below shows the processing order of BGP policy filters.
Table 16  BGP Policy Processing Order

<table>
<thead>
<tr>
<th>Inbound</th>
<th>Outbound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route map</td>
<td>Distribute list</td>
</tr>
<tr>
<td>Filter list, AS-path access list, or IP policy</td>
<td>IP prefix list</td>
</tr>
<tr>
<td>IP prefix list</td>
<td>Filter list, AS-path access list, or IP policy</td>
</tr>
<tr>
<td>Distribute list</td>
<td>Route map</td>
</tr>
</tbody>
</table>

Whenever there is a change in the routing policy due to a configuration change, BGP peering sessions must be reset using the `clear ip bgp` command. Cisco IOS XE software supports the following three mechanisms to reset BGP peering sessions:

- **Hard reset**—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer.
- **Soft reset**—A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reset uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply a new BGP policy without disrupting the network. Soft reset can be configured for inbound or outbound sessions.
- **Dynamic inbound soft reset**—The route refresh capability, as defined in RFC 2918, allows the local router to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for nondisruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP routers must support the route refresh capability.

To determine if a BGP router supports this capability, use the `show ip bgp neighbors` command. The following message is displayed in the output when the router supports the route refresh capability:

```
Received route refresh capability from peer.
```

**BGP Communities**

BGP communities are used to group routes (also referred to as color routes) that share common properties, regardless of network, autonomous system, or any physical boundaries. In large networks applying a common routing policy through prefix-lists or access-lists requires individual peer statements on each networking device. Using the BGP community attribute BGP speakers, with common routing policies, can implement inbound or outbound route filters based on the community tag rather than consult large lists of individual permit or deny statements.

Standard community lists are used to configure well-known communities and specific community numbers. Expanded community lists are used to filter communities using a regular expression. Regular expressions are used to configure patterns to match community attributes.

The community attribute is optional, which means that it will not be passed on by networking devices that do not understand communities. Networking devices that understand communities must be configured to handle the communities or they will be discarded.

There are four predefined communities:

- **no-export**—Do not advertise to external BGP peers.
- **no-advertise**—Do not advertise this route to any peer.
Extended Communities

Extended community attributes are used to configure, filter, and identify routes for virtual routing and forwarding (VRF) instances and Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers. All regular expression configuration options are supported. The route target (RT) and site of origin (SoO) extended community attributes are supported by the standard range of extended community lists.

Route Target Extended Community Attribute

The RT extended community attribute is configured with the `rt` keyword of the `ip extcommunity-list` command. This attribute is used to identify a set of sites and VRFs that may receive routes that are tagged with the configured route target. Configuring the route target extended community attribute with a route allows that route to be placed in the per-site forwarding tables that are used for routing traffic that is received from corresponding sites.

Site of Origin Extended Community Attribute

The SoO extended community attribute is configured with the `soo` keyword of the `ip extcommunity-list` command. This attribute uniquely identifies the site from which the provider edge (PE) router learned the route. All routes learned from a particular site must be assigned the same SoO extended community attribute, regardless if a site is connected to a single PE router or multiple PE routers. Configuring this attribute prevents routing loops from occurring when a site is multihomed. The SoO extended community attribute is configured on the interface and is propagated into BGP through redistribution. The SoO extended community attribute can be applied to routes that are learned from VRFs. The SoO extended community attribute should not be configured for stub sites or sites that are not multihomed.

IP Extended Community-List Configuration Mode

Named and numbered extended community lists can be configured in IP extended community-list configuration mode. The IP extended community-list configuration mode supports all of the functions that are available in global configuration mode. In addition, the following operations can be performed:

- Configure sequence numbers for extended community list entries.
- Resequence existing sequence numbers for extended community list entries.
- Configure an extended community list to use default values.

Both standard and expanded community lists have a limitation of 100 community groups that can be configured within each type of list. A named community list does not have this limitation.
Default Sequence Numbering

Extended community list entries start with the number 10 and increment by 10 for each subsequent entry when no sequence number is specified, when default behavior is configured, and when an extended community list is resequenced without specifying the first entry number or the increment range for subsequent entries.

Resequencing Extended Community Lists

Extended community-list entries are sequenced and resequenced on a per-extended community list basis. The resequence command can be used without any arguments to set all entries in a list to default sequence numbering. The resequence command also allows the sequence number of the first entry and increment range to be set for each subsequent entry. The range of configurable sequence numbers is from 1 to 2147483647.

Extended Community Lists

Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The ip extcommunity-list command is used to configure named or numbered extended community lists. All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

Administrative Distance

Administrative distance is a measure of the preference of different routing protocols. BGP has a distance bgp command that allows you to set different administrative distances for three route types: external, internal, and local. BGP, like other protocols, prefers the route with the lowest administrative distance.

BGP Route Map Policy Lists

BGP route map policy lists allow a network operator to group route map match clauses into named lists called policy lists. A policy list functions like a macro. When a policy list is referenced in a route map, all of the match clauses are evaluated and processed as if they had been configured directly in the route map. This enhancement simplifies the configuration of BGP routing policy in medium-size and large networks because a network operator can preconfigure policy lists with groups of match clauses and then reference these policy lists within different route maps. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.

A policy list functions like a macro when it is configured in a route map and has the following capabilities and characteristics:

- When a policy list is referenced within a route map, all the match statements within the policy list are evaluated and processed.
- Two or more policy lists can be configured with a route map. Policy lists can be configured within a route map to be evaluated with AND or OR semantics.
- Policy lists can coexist with any other preexisting match and set statements that are configured within the same route map but outside of the policy lists.
- When multiple policy lists perform matching within a route map entry, all policy lists match on the incoming attribute only.

Policy lists support only match clauses and do not support set clauses. Policy lists can be configured for all applications of route maps, including redistribution, and can also coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists.
BGP Route Map with a Continue Clause

In Cisco IOS XE Release 2.1 and later releases, the continue clause was introduced into BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduced the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. Before the continue clause was introduced, route map configuration was linear and did not allow any control over the flow of a route map.

In Cisco IOS XE Release 2.1 and later releases, support for continue clauses for outbound route maps was introduced.

- Route Map Operation Without Continue Clauses, page 176
- Route Map Operation with Continue Clauses, page 176
- Match Operations with Continue Clauses, page 176
- Set Operations with Continue Clauses, page 177

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map “falls through” and evaluates the next sequence number of the route map until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order starting with the lowest sequence number and ending with the highest sequence number.

If the route map contains only set clauses, the set clauses will be executed automatically, and the route map will not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map will continue to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (or jump to) a specific route map entry by specifying the sequence number, or if a sequence number is not specified, the continue clause will go to the next sequence number. This behavior is called an “implied continue.” If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

Match Operations with Continue Clauses

If a match clause does not exist in the route map entry but a continue clause does, the continue clause will be automatically executed and go to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map entry contains a continue clause, the route map will execute the continue clause if a successful match occurs. If a continue clause does not exist in the next route map entry, the route map will be evaluated normally. If a continue clause exists in the next route map entry but a
match does not occur, the route map will not continue and will “fall through” to the next sequence number if one exists.

**Set Operations with Continue Clauses**

Set clauses are saved during the match clause evaluation process and executed after the route-map evaluation is completed. The set clauses are evaluated and executed in the order in which they were configured. Set clauses are executed only after a successful match occurs, unless the route map does not contain a match clause. The continue statement proceeds to the specified route map entry only after configured set actions are performed. If a set action occurs in the first route map and then the same set action occurs again, with a different value, in a subsequent route map entry, the last set action may override any previous set actions that were configured with the same set command unless the set command permits more than one value. For example, the `set as-path prepend` command permits more than one autonomous system number to be configured.

---

**Note**

A continue clause can be executed, without a successful match, if a route map entry does not contain a match clause.

---

**Note**

Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. For an example, see "Filtering Traffic Using Continue Clauses in a BGP Route Map Examples".

### How to Connect to a Service Provider Using External BGP

- [Influencing Inbound Path Selection](#), page 177
- [Influencing Outbound Path Selection](#), page 186
- [Configuring BGP Peering with ISPs](#), page 193
- [Configuring BGP Policies](#), page 208

### Influencing Inbound Path Selection

BGP can be used to influence the choice of paths in another autonomous system. There may be several reasons for wanting BGP to choose a path that is not the obvious best route, for example, to avoid some types of transit traffic passing through an autonomous system or perhaps to avoid a very slow or congested link. BGP can influence inbound path selection using one of the following BGP attributes:

- AS-path
- MED

Perform one of the following tasks to influence inbound path selection:

- [Influencing Inbound Path Selection by Modifying the AS-path Attribute](#), page 178
- [Influencing Inbound Path Selection by Setting the MED Attribute](#), page 182
Influencing Inbound Path Selection by Modifying the AS-path Attribute

One of the methods that BGP can use to influence the choice of paths in another autonomous system is to modify the AS-path attribute. For example, in the figure below, Router A advertises its own network, 172.17.1.0, to its BGP peers in autonomous system 45000 and autonomous system 60000. When the routing information is propagated to autonomous system 50000, the routers in autonomous system 50000 have network reachability information about network 172.17.1.0 from two different routes. The first route is from autonomous system 45000 with an AS-path consisting of 45000, 40000, the second route is through autonomous system 55000 with an AS-path of 55000, 60000, 40000. If all other BGP attribute values are the same, Router C in autonomous system 50000 would choose the route through autonomous system 45000 for traffic destined for network 172.17.1.0 because it is the shortest route in terms of autonomous systems traversed.

Autonomous system 40000 now receives all traffic from autonomous system 50000 for the 172.17.1.0 network through autonomous system 45000. If, however, the link between autonomous system 45000 and autonomous system 40000 is a really slow and congested link, the `set as-path prepend` command can be used at Router A to influence inbound path selection for the 172.17.1.0 network by making the route through autonomous system 45000 appear to be longer than the path through autonomous system 60000. The configuration is done at Router A in the figure below by applying a route map to the outbound BGP updates to Router B. Using the `set as-path prepend` command, all the outbound BGP updates from Router A to Router B will have their AS-path attribute modified to add the local autonomous system number 40000 twice. After the configuration, autonomous system 50000 receives updates about the 172.17.1.0 network through autonomous system 45000. The new AS-path is 45000, 40000, 40000, and 40000, which is now longer than the AS-path from autonomous system 55000 (unchanged at a value of 55000, 60000, 40000). Networking devices in autonomous system 50000 will now prefer the route through autonomous system 55000 to forward packets with a destination address in the 172.17.1.0 network.

![Network Topology for Modifying the AS-path Attribute](image)

---

**Figure 17** Network Topology for Modifying the AS-path Attribute

---

Connecting to a Service Provider Using External BGP

Influencing Inbound Path Selection by Modifying the AS-path Attribute

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2
Perform this task to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. The configuration is performed at Router A in the figure below.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [unicast | multicast | vrf vrf-name]`
5. `network network-number [mask network-mask][route-map route-map-name]`
6. `neighbor {ip-address | peer-group-name} remote-as autonomous-system-number`
7. `neighbor {ip-address | peer-group-name} route-map map-name {in | out}`
8. `neighbor {ip-address | peer-group-name} activate`
9. `exit`
10. `exit`
11. `route-map map-name [permit | deny] [sequence-number]`
12. `set as-path {tag | prepend as-path-string}`
13. `end`
14. `show running-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 40000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

| Step 4 | address-family ipv4 [unicast | multicast] vrf vrf-name |
|--------|-------------------------------------------------------------|
| **Example:** | Router(config-router)# address-family ipv4 unicast |

**Purpose:**

Specifies the IPv4 address family and enters address family configuration mode.

- The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.
- The **multicast** keyword specifies IPv4 multicast address prefixes.
- The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>network network-number [mask network-mask] [route-map route-map-name]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</td>
</tr>
</tbody>
</table>

**Purpose:**

Specifies a network as local to this autonomous system and adds it to the BGP routing table.

- For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

| Step 6 | neighbor {ip-address | peer-group-name} remote-as autonomous-system-number |
|--------|-----------------------------------------------------------------------|
| **Example:** | Router(config-router-af)# neighbor 192.168.1.2 remote-as 45000 |

**Purpose:**

Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

- In this example, the BGP peer on Router B at 192.168.1.2 is added to the IPv4 multiprotocol BGP neighbor table and will receive BGP updates.

| Step 7 | neighbor {ip-address | peer-group-name} route-map map-name [in | out] |
|--------|-----------------------------------------------------------------------|
| **Example:** | Router(config-router-af)# neighbor 192.168.1.2 route-map PREPEND out |

**Purpose:**

Applies a route map to incoming or outgoing routes.

- In this example, the route map named PREPEND is applied to outbound routes to Router B.

| Step 8 | neighbor {ip-address | peer-group-name} activate |
|--------|------------------------------------------------------------------------------------------------|
| **Example:** | Router(config-router-af)# neighbor 192.168.1.2 activate |

**Purpose:**

Enables address exchange for address family IPv4 unicast for the BGP neighbor at 192.168.1.2 on Router B.
## Command or Action | Purpose
--- | ---
**Step 9** exit | Exits address family configuration mode and enters router configuration mode.

**Example:**

Router(config-router-af)# exit

**Step 10** exit | Exits router configuration mode and enters global configuration mode.

**Example:**

Router(config-router)# exit

**Step 11** route-map map-name [permit| deny] [sequence-number] | Configures a route map and enters route map configuration mode.

- In this example, a route map named PREPEND is created and if there is a subsequent matching of criteria.

**Example:**

Router(config)# route-map PREPEND permit 10

**Step 12** set as-path {tag | prepend as-path-string} | Modifies an autonomous system path for BGP routes.

- Use the **prepend** keyword to "prepend" an arbitrary autonomous system path string to BGP routes. Usually the local autonomous system number is prepended multiple times, increasing the autonomous system path length.
- In this example, two additional autonomous system entries are added to the autonomous system path for outbound routes to Router B.

**Example:**

Router(config-route-map)# set as-path prepend 40000 40000

**Step 13** end | Exits route map configuration mode and returns to privileged EXEC mode.

**Example:**

Router(config-route-map)# end

**Step 14** show running-config | Displays the running configuration file.

**Example:**

Router# show running-config
Influencing Inbound Path Selection by Setting the MED Attribute

One of the methods that BGP can use to influence the choice of paths into another autonomous system is to set the MED attribute. The MED attribute indicates (to an external peer) a preferred path to an autonomous system. If there are multiple entry points to an autonomous system, the MED can be used to influence another autonomous system to choose one particular entry point. A metric is assigned using route maps where a lower MED metric is preferred by the software over a higher MED metric.

Perform this task to influence inbound path selection by setting the MED metric attribute. The configuration is performed at Router B and Router D in the figure below. Router B advertises the network 172.16.1.0. to its BGP peer, Router E in autonomous system 50000. Using a simple route map Router B sets the MED metric to 50 for outbound updates. The task is repeated at Router D but the MED metric is set to 120. When Router E receives the updates from both Router B and Router D the MED metric is stored in the BGP routing table. Before forwarding packets to network 172.16.1.0, Router E compares the attributes from peers in the same autonomous system (both Router B and Router D are in autonomous...
system 45000). The MED metric for Router B is less than the MED for Router D, so Router E will forward the packets through Router B.

**Figure 18  Network Topology for Setting the MED Attribute**

Use the `bgp always-compare-med` command to compare MED attributes from peers in other autonomous systems.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **router bgp** `autonomous-system-number`
4. **neighbor** `{ip-address | peer-group-name}` **remote-as** `autonomous-system-number`
5. **address-family** `ipv4` [unicast | multicast] `vrf` `vrf-name`
6. **network** `network-number` [mask `network-mask`] [route-map `route-map-name`]
7. **neighbor** `{ip-address | peer-group-name}` **route-map** `map-name` {in | out}
8. **exit**
9. **exit**
10. **route-map** `map-name` [permit] deny [sequence-number]
11. **set** metric `value`
12. **end**
13. Repeat Step 1 through Step 12 at Router D.
14. **show ip bgp** [network] [network-mask]
## Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.3.2 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family ipv4 {unicast</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
</tbody>
</table>

### Connecting to a Service Provider Using External BGP

Influencing Inbound Path Selection by Setting the MED Attribute

---

*IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2*
### Command or Action

<table>
<thead>
<tr>
<th>Step 6</th>
<th>network network-number [mask network-mask] [route-map route-map-name]</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• For exterior protocols the <code>network</code> command controls which networks are advertised. Interior protocols use the <code>network</code> command to determine where to send updates.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# network 172.16.1.0 mask 255.255.255.0
```

| Step 7 | neighbor {ip-address | peer-group-name} route-map map-name {in | out} | Purpose |
|--------|--------------------------------------------------------|---------|
|        | Applies a route map to incoming or outgoing routes. |
|        | • In this example, the route map named MED is applied to outbound routes to the BGP peer at Router E. |

**Example:**

```
Router(config-router-af)# neighbor 192.168.3.2 route-map MED out
```

<table>
<thead>
<tr>
<th>Step 8</th>
<th>exit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits address family configuration mode and enters router configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# exit
```

<table>
<thead>
<tr>
<th>Step 9</th>
<th>exit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits router configuration mode and enters global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# exit
```

<table>
<thead>
<tr>
<th>Step 10</th>
<th>route-map map-name [permit] [deny] [sequence-number]</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configures a route map and enters route map configuration mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• In this example, a route map named MED is created.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config)# route-map MED permit 10
```

<table>
<thead>
<tr>
<th>Step 11</th>
<th>set metric value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sets the MED metric value.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-route-map)# set metric 50
```

<table>
<thead>
<tr>
<th>Step 12</th>
<th>end</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits route map configuration mode and enters privileged EXEC mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-route-map)# end
```
### Command or Action

<table>
<thead>
<tr>
<th>Step 13</th>
<th>Repeat Step 1 through Step 12 at Router D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 14</td>
<td><code>show ip bgp [network] [network-mask]</code></td>
</tr>
</tbody>
</table>

**Example:**

```
Router# show ip bgp 172.17.1.0 255.255.255.0
```

**Purpose**

(Optional) Displays the entries in the BGP routing table.

- Use this command at Router E in the figure above when both Router B and Router D have configured the MED attribute.
- Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

---

**Examples**

The following output is from Router E in the figure below after this task has been performed at both Router B and Router D. Note the metric (MED) values for the two routes to network 172.16.1.0. The peer 192.168.2.1 at Router D has a metric of 120 for the path to network 172.16.1.0 whereas the peer 192.168.3.1 at Router B has a metric of 50. The entry for the peer 192.168.3.1 at Router B has the word best at the end of the entry to show that Router E will choose to send packets destined for network 172.16.1.0 via Router B because the MED metric is lower.

```
Router# show ip bgp 172.16.1.0
BGP routing table entry for 172.16.1.0/24, version 10
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Advertised to update-groups:
  1
  45000
  192.168.2.1 from 192.168.2.1 (192.168.2.1)
    Origin IGP, metric 120, localpref 100, valid, external
  45000
  192.168.3.1 from 192.168.3.1 (172.17.1.99)
    Origin IGP, metric 50, localpref 100, valid, external, best
```

---

**Influencing Outbound Path Selection**

BGP can be used to influence the choice of paths for outbound traffic from the local autonomous system. This section contains two methods that BGP can use to influence outbound path selection:

- Using the Local_Pref attribute
- Using the BGP outbound route filter (ORF) capability

Perform one of the following tasks to influence outbound path selection:

- **Influencing Outbound Path Selection Using the Local_Pref Attribute**, page 186
- **Filtering Outbound BGP Route Prefixes**, page 189

---

**Influencing Outbound Path Selection Using the Local_Pref Attribute**

One of the methods to influence outbound path selection is to use the BGP Local-Pref attribute. Perform this task using the local preference attribute to influence outbound path selection. If there are several paths to the same destination the local preference attribute with the highest value indicates the preferred path.

Refer to the figure below for the network topology used in this task. Both Router B and Router C are configured. Autonomous system 45000 receives updates for network 192.168.3.0 via autonomous system 40000 and autonomous system 50000. Router B is configured to set the local preference value to 150 for all updates to autonomous system 40000. Router C is configured to set the local preference value for all
updates to autonomous system 50000 to 200. After the configuration, local preference information is exchanged within autonomous system 45000. Router B and Router C now see that updates for network 192.168.3.0 have a higher preference value from autonomous system 50000 so all traffic in autonomous system 45000 with a destination network of 192.168.3.0 is sent out via Router C.

**Figure 19**  
**Network Topology for Outbound Path Selection**

---

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast | vrf vrf-name]
5. network network-number [mask network-mask] [route-map route-map-name]
6. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
7. bgp default local-preference value
8. neighbor {ip-address | peer-group-name} activate
9. end
10. Repeat Step 1 through Step 9 at Router C but change the IP address of the peer, the autonomous system number, and set the local preference value to 200.
11. show ip bgp [network] [network-mask]
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [unicast</td>
<td>multicast] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> network network-number [mask network-mask] [route-map route-map-name]</td>
<td>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.1.2 remote-as 40000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 7**  
`bgp default local-preference value`  
Example:

```
Router(config-router-af)# bgp default local-preference 150
```

Changes the default local preference value.
- In this example, the local preference is changed to 150 for all updates from autonomous system 40000 to autonomous system 45000.
- By default, the local preference value is 100.

**Step 8**  
`neighbor {ip-address|peer-group-name} activate`  
Example:

```
Router(config-router-af)# neighbor 192.168.1.2 activate
```

Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

**Step 9**  
`end`  
Example:

```
Router(config-router-af)# end
```

Exits route map configuration mode and enters privileged EXEC mode.

**Step 10**  
Repeat Step 1 through Step 9 at Router C but change the IP address of the peer, the autonomous system number, and set the local preference value to 200.

**Step 11**  
`show ip bgp [network] [network-mask]`  
Example:

```
Router# show ip bgp 192.168.3.0 255.255.255.0
```

Displays the entries in the BGP routing table.
- Enter this command at both Router B and Router C and note the Local_Pref value. The route with the highest preference value will be the preferred route to network 192.168.3.0.

**Note**  
Only the syntax applicable to this task is used in this example. For more details, see the `Cisco IOS IP Routing: BGP Command Reference`.

### Filtering Outbound BGP Route Prefixes

Perform this task to use BGP prefix-based outbound route filtering to influence outbound path selection.
- BGP Prefix-Based Outbound Route Filtering, page 189

**BGP Prefix-Based Outbound Route Filtering**  
BGP prefix-based outbound route filtering uses the BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers.Configuring BGP ORF can help reduce the amount of system resources required for generating and processing routing updates by filtering out unwanted routing updates at the source. For example, BGP ORF can be used to reduce the amount of processing required on a router that is not accepting full routes from a service provider network.
The BGP prefix-based outbound route filtering is enabled through the advertisement of ORF capabilities to peer routers. The advertisement of the ORF capability indicates that a BGP peer will accept a prefix list from a neighbor and apply the prefix list to locally configured ORFs (if any exist). When this capability is enabled, the BGP speaker can install the inbound prefix list filter to the remote peer as an outbound filter, which reduces unwanted routing updates.

The BGP prefix-based outbound route filtering can be configured with send or receive ORF capabilities. The local peer advertises the ORF capability in send mode. The remote peer receives the ORF capability in receive mode and applies the filter as an outbound policy. The local and remote peers exchange updates to maintain the ORF on each router. Updates are exchanged between peer routers by address family depending on the ORF prefix list capability that is advertised. The remote peer starts sending updates to the local peer after a route refresh has been requested with the clear ip bgp in prefix-filter command or after an ORF prefix list with immediate status is processed. The BGP peer will continue to apply the inbound prefix list to received updates after the local peer pushes the inbound prefix list to the remote peer.

BGP peering sessions must be established, and BGP ORF capabilities must be enabled on each participating router before prefix-based ORF announcements can be received.

---

**Note**

- BGP prefix-based outbound route filtering does not support multicast.
- IP addresses that are used for outbound route filtering must be defined in an IP prefix list. BGP distribute lists and IP access lists are not supported.
- Outbound route filtering is configured on only a per-address family basis and cannot be configured under the general session or BGP routing process.
- Outbound route filtering is configured for external peering sessions only.

---

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip prefix-list list-name [seq seq-value] {deny network | length | permit network | length} [ge ge-value] [le le-value]`
4. `router bgp autonomous-system-number`
5. `address-family ipv4 [unicast | multicast | vrf vrf-name]`
6. `neighbor {ip-address | peer-group-name} remote-as autonomous-system-number`
7. `neighbor ip-address ebgp-multihop [hop-count]`
8. `neighbor ip-address capability orf prefix-list [send | receive | both]`
9. `neighbor {ip-address | peer-group-name} prefix-list prefix-list-name [in | out]`
10. `end`
11. `clear ip bgp {ip-address | *} in prefix-filter`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Example:**  
Router> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:**  
Router# configure terminal | |
| **Step 3** ip prefix-list list-name [seq seq-value]  
  {deny network / length | permit network / length}[ge ge-value] [le le-value] | Creates a prefix list for prefix-based outbound route filtering.  
  • Outbound route filtering supports prefix length matching, wildcard-based prefix matching, and exact address prefix matching on a per address-family basis.  
  • The prefix list is created to define the outbound route filter. The filter must be created when the outbound route filtering capability is configured to be advertised in send mode or both mode. It is not required when a peer is configured to advertise receive mode only.  
  • The example creates a prefix list named FILTER that defines the 192.168.1.0/24 subnet for outbound route filtering. |
| **Example:**  
Router(config)# ip prefix-list FILTER seq 10 permit 192.168.1.0/24 | |
| **Step 4** router bgp autonomous-system-number | Enters router configuration mode, and creates a BGP routing process. |
| **Example:**  
Router(config)# router bgp 100 | |
| **Step 5** address-family ipv4 [unicast | multicast]  
  vrf vrf-name | Specifies the IPv4 address family and enters address family configuration mode.  
  • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.  
  • The **multicast** keyword specifies IPv4 multicast address prefixes.  
  • The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. |
| **Example:**  
Router(config-router)# address-family ipv4 unicast | Note  
Outbound route filtering is configured on a per-address family basis. |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 6** neighbor {ip-address| peer-group-name} remote-as autonomous-system-number | Establishes peering with the specified neighbor or peer group. BGP peering must be established before ORF capabilities can be exchanged.  
  - The example establishes peering with the 10.1.1.1 neighbor. |

**Example:**

```
Router(config-router-af)# neighbor 10.1.1.1 remote-as 200
```

**Step 7** neighbor ip-address ebgp-multihop [hop-count] | Accepts or initiates BGP connections to external peers residing on networks that are not directly connected. |

**Example:**

```
Router(config-router-af)# neighbor 10.1.1.1 ebgp-multihop
```

**Step 8** neighbor ip-address capability orf prefix-list [send | receive | both] | Enables the ORF capability on the local router, and enables ORF capability advertisement to the BGP peer specified with the ip-address argument.  
  - The send keyword configures a router to advertise ORF send capabilities.  
  - The receive keyword configures a router to advertise ORF receive capabilities.  
  - The both keyword configures a router to advertise send and receive capabilities.  
  - The remote peer must be configured to either send or receive ORF capabilities before outbound route filtering is enabled.  
  - The example configures the router to advertise send and receive capabilities to the 10.1.1.1 neighbor. |

**Example:**

```
Router(config-router-af)# neighbor 10.1.1.1 capability orf prefix-list both
```

**Step 9** neighbor {ip-address| peer-group-name} prefix-list prefix-list-name {in | out} | Applies an inbound prefix-list filter to prevent distribution of BGP neighbor information.  
  - In this example, the prefix list named FILTER is applied to incoming advertisements from the 10.1.1.1 neighbor, which prevents distribution of the 192.168.1.0/24 subnet. |

**Example:**

```
Router(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in
```

**Step 10** end | Exits address family configuration mode, and enters privileged EXEC mode. |

**Example:**

```
Router(config-router-af)# end
```
**Command or Action**

| Step 11 clear ip bgp {ip-address | *} in prefix-filter |
|----------------------------------------------------------|

**Purpose**

Clears BGP outbound route filters and initiates an inbound soft reset.

- A single neighbor or all neighbors can be specified.

**Example:**

```
Router# clear ip bgp 10.1.1.1 in prefix-filter
```

**Note**

The inbound soft refresh must be initiated with the `clear ip bgp` command in order for this feature to function.

---

**Configuring BGP Peering with ISPs**

BGP was developed as an interdomain routing protocol and connecting to ISPs is one of the main functions of BGP. Depending on the size of your network and the purpose of your business, there are many different ways to connect to your ISP. Multihoming to one or more ISPs provides redundancy in case an external link to an ISP fails. This section introduces some optional tasks that can be used to connect to a service provider using multihoming techniques. Smaller companies may use just one ISP but require a backup route to the ISP. Larger companies may have access to two ISPs, using one of the connections as a backup, or may need to configure a transit autonomous system.

Perform one of the following optional tasks to connect to one or more ISPs:

- Configuring Multihoming with Two ISPs, page 193
- Multihoming with a Single ISP, page 197
- Configuring Multihoming to Receive the Full Internet Routing Table, page 204

**Configuring Multihoming with Two ISPs**

Perform this task to configure your network to access two ISPs, where one ISP is the preferred route and the second ISP is a backup route. In the figure below Router B in autonomous system 45000 has BGP peers in two ISPs, autonomous system 40000 and autonomous system 50000. Using this task, Router B will be configured to prefer the route to the BGP peer at Router A in autonomous system 40000.

All routes learned from this neighbor will have an assigned weight. The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.
The weights assigned with the `set weight` route-map configuration command override the weights assigned using the `neighbor weight` command.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [unicast | multicast | vrf vrf-name]`
5. `network network-number [mask network-mask]`
6. `neighbor {ip-address|peer-group-name} remote-as autonomous-system-number`
7. `neighbor {ip-address|peer-group-name} weight number`
8. `neighbor {ip-address|peer-group-name} remote-as autonomous-system-number`
9. `neighbor {ip-address|peer-group-name} weight number`
10. `end`
11. `clear ip bgp [*|ip-address|peer-group-name] [soft [in | out]]`
12. `show ip bgp [network] [network-mask]`
### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable          | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| Example:                   |         |
| Router> enable             |         |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example:                   |         |
| Router# configure terminal |         |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode, and creates a BGP routing process. |
| Example:                   |         |
| Router(config)# router bgp 45000 |         |
| **Step 4** address-family ipv4 [unicast | multicast] vrf vrf-name] | Specifies the IPv4 address family and enters address family configuration mode.  
  • The **unicast** keyword specifies the IPv4 unicast address family.  
  By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.  
  • The **multicast** keyword specifies IPv4 multicast address prefixes.  
  • The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. |
| Example:                   |         |
| Router(config-router)# address-family ipv4 unicast |         |
| **Step 5** network network-number [mask network-mask] | Specifies a network as local to this autonomous system and adds it to the BGP routing table.  
  • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates. |
| Example:                   |         |
| Router(config-router-asf)# network 172.17.1.0 mask 255.255.255.0 |         |
| **Step 6** neighbor [ip-address| peer-group-name] remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. |
| Example:                   |         |
| Router(config-router-asf)# neighbor 192.168.1.2 remote-as 40000 |         |
### Configuring Multihoming with Two ISPs

#### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} weight number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In this example, the weight attribute for routes received from the BGP peer 192.168.1.2 is set to 150.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 192.168.1.2 weight 150</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 192.168.3.2 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} weight number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In this example, the weight attribute for routes received from the BGP peer 192.168.3.2 is set to 100.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 192.168.3.2 weight 100</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>end</td>
<td>Exits address family configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td>clear ip bgp {*</td>
<td>ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# clear ip bgp *</td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td>show ip bgp [network] [network-mask]</td>
<td>Displays the entries in the BGP routing table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>• Enter this command at Router B to see the weight attribute for each route to a BGP peer. The route with the highest weight attribute will be the preferred route to network 172.17.1.0.</td>
</tr>
</tbody>
</table>

**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.  

---

**Connecting to a Service Provider Using External BGP**

---

**IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2**
Examples

The following example shows the BGP routing table at Router B with the weight attributes assigned to routes. The route through 192.168.3.2 (Router E in the figure above) has the highest weight attribute and will be the preferred route to network 172.17.1.0.

BGP table version is 8, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.0/24</td>
<td>192.168.1.2</td>
<td>0</td>
<td>100</td>
<td>40000</td>
<td>i</td>
</tr>
<tr>
<td>10.2.2.0/24</td>
<td>192.168.3.2</td>
<td>0</td>
<td>150</td>
<td>50000</td>
<td>i</td>
</tr>
<tr>
<td>172.17.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Multihoming with a Single ISP

Perform this task to configure your network to access one of two connections to a single ISP, where one of the connections is the preferred route and the second connection is a backup route. In the figure above Router E in autonomous system 50000 has two BGP peers in a single autonomous system, autonomous system 45000. Using this task, autonomous system 50000 does not learn any routes from autonomous system 45000 and is sending its own routes using BGP. This task is configured at Router E in the figure above and covers three features about multihoming to a single ISP:

- **Outbound traffic**—Router E will forward default routes and traffic to autonomous system 45000 with Router B as the primary link and Router D as the backup link. Static routes are configured to both Router B and Router D with a lower distance configured for the link to Router B.
- **Inbound traffic**—Inbound traffic from autonomous system 45000 is configured to be sent from Router B unless the link fails when the backup route is to send traffic from Router D. To achieve this, outbound filters are set using the MED metric.
- **Prevention of transit traffic**—A route map is configured at Router E in autonomous system 50000 to block all incoming BGP routing updates to prevent autonomous system 50000 from receiving transit traffic from the ISP in autonomous system 45000.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. network network-number [mask network-mask][route-map route-map-name]
5. address-family ipv4 [unicast | multicast| vrf vrf-name]
6. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
7. neighbor {ip-address | peer-group-name} route-map map-name{in | out}
8. Repeat Step 7 to apply another route map to the neighbor specified in Step 7.
9. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
10. neighbor {ip-address | peer-group-name} route-map map-name{in | out}
11. Repeat Step 10 to apply another route map to the neighbor specified in Step 10.
12. exit
13. exit
14. ip route prefix mask [ip-address | interface-type interface-number[ip-address]] [distance] [name] [permanent] [track number][tag tag]
15. Repeat Step 14 to establish another static route.
16. route-map map-name [permit| deny][sequence-number]
17. set metric value
18. exit
19. route-map map-name [permit| deny][sequence-number]
20. set metric value
21. exit
22. route-map map-name [permit| deny][sequence-number]
23. end
24. show ip route [ip-address] [mask] [longer-prefixes]
25. show ip bgp [network] [network-mask]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> network network-number [mask network-mask] [route-map route-map-name]</td>
<td>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# network 10.2.2.0 mask 255.255.255.0</td>
<td>• For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.</td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family ipv4 [unicast</td>
<td>multicast] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td>• The unicast keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the unicast keyword is not specified with the address-family ipv4 command.</td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.2.1 remote-as 45000</td>
<td>• In this example, the BGP peer at Router D is added to the BGP routing table.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} route-map map-name [in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the first example, the route map named BLOCK is applied to inbound routes at Router E.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the second example, the route map named SETMETRIC1 is applied to outbound routes to Router D.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note Two examples are shown here because the task example requires both these statements to be configured.</td>
</tr>
<tr>
<td>8</td>
<td>Repeat Step 7 to apply another route map to the neighbor specified in Step 7.</td>
<td>--</td>
</tr>
<tr>
<td>9</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In this example, the BGP peer at Router D is added to the BGP routing table.</td>
</tr>
<tr>
<td>10</td>
<td>neighbor {ip-address</td>
<td>peer-group-name} route-map map-name [in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the first example, the route map named BLOCK is applied to inbound routes at Router E.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the second example, the route map named SETMETRIC2 is applied to outbound routes to Router D.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note Two examples are shown here because the task example requires both these statements to be configured.</td>
</tr>
<tr>
<td>11</td>
<td>Repeat Step 10 to apply another route map to the neighbor specified in Step 10.</td>
<td>--</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> exit</td>
<td>Exits address family configuration mode and enters router configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# exit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **Step 14** `ip route prefix mask [ip-address | interface-type interface-number][ip-address]] [distance] [name] [permanent] [track number][tag tag]` | Establishes a static route.  
- In the first example, a static route to BGP peer 192.168.2.1 is established and given an administrative distance of 50.  
- In the second example, a static route to BGP peer 192.168.3.1 is established and given an administrative distance of 40. The lower administrative distance makes this route via Router B the preferred route. |
| **Example:** | |
| Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50 | |
| **Example:** | |
| Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50 | |
| **Example:** | |
| and | |
| **Example:** | |
| Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1 40 | |
| **Step 15** Repeat Step 14 to establish another static route. | -- |
| **Step 16** `route-map map-name [permit| deny][sequence-number]` | Configures a route map and enters route map configuration mode.  
- In this example, a route map named SETMETRIC1 is created. |
<p>| <strong>Example:</strong> | |
| Router(config)# route-map SETMETRIC1 permit 10 | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 17</strong> set metric value</td>
<td>Sets the MED metric value.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# set metric 100</td>
</tr>
<tr>
<td><strong>Step 18</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# exit</td>
</tr>
<tr>
<td><strong>Step 19</strong> route-map map-name [permit</td>
<td>deny][sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# route-map SETMETRIC2 permit 10</td>
</tr>
<tr>
<td><strong>Step 20</strong> set metric value</td>
<td>Sets the MED metric value.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# set metric 50</td>
</tr>
<tr>
<td><strong>Step 21</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# exit</td>
</tr>
<tr>
<td><strong>Step 22</strong> route-map map-name [permit</td>
<td>deny][sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# route-map BLOCK deny 10</td>
</tr>
<tr>
<td><strong>Step 23</strong> end</td>
<td>Exits route map configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# end</td>
</tr>
</tbody>
</table>
**Step 24**  show ip route [ip-address] [mask] [longer-prefixes]

**Example:**

Router# show ip route

**Step 25**  show ip bgp [network] [network-mask]

**Example:**

Router# show ip bgp 172.17.1.0 255.255.255.0

---

**Examples**

The following example shows output from the `show ip route` command entered at Router E after this task has been configured and Router B and Router D have received update information containing the MED metric. Note that the gateway of last resort is set as 192.168.3.1, which is the route to Router B.

```
Router# show ip route
Codes: 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
      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
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 192.168.3.1 to network 0.0.0.0
10.0.0.0/24 is subnetted, 1 subnets
   C       10.2.2.0 is directly connected, GigabitEthernet0/0/0
   C    192.168.2.0/24 is directly connected, Serial3/0/0
   C    192.168.3.0/24 is directly connected, Serial2/0/0
   S*   0.0.0.0/0 [40/0] via 192.168.3.1
```

The following example shows output from the `show ip bgp` command entered at Router E after this task has been configured and Router B and Router D have received routing updates. The route map BLOCK has denied all routes coming in from autonomous system 45000 so the only network shown is the local network.

```
Router# show ip bgp
BGP table version is 2, local router ID is 10.2.2.99
Status codes: s suppressed, d damped, h history, v valid, b best, i internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network  Next Hop       Metric LocPrf Weight Path
*0 10.2.2.0/24 0.0.0.0 0 [40/0] via 192.168.3.1
```

The following example shows output from the `show ip bgp` command entered at Router B after this task has been configured at Router E and Router B has received routing updates. Note the metric of 50 for network 10.2.2.0.

```
Router# show ip bgp
```
BGP table version is 7, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.1.1.0/24</td>
<td>192.168.1.2</td>
<td>0</td>
<td>0</td>
<td>40000</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 10.2.2.0/24</td>
<td>192.168.3.2</td>
<td>50</td>
<td>0</td>
<td>50000</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 172.16.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>*&gt; 172.17.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

The following example shows output from the `show ip bgp` command entered at Router D after this task has been configured at Router E and Router D has received routing updates. Note the metric of 100 for network 10.2.2.0.

Router# show ip bgp
BGP table version is 3, local router ID is 192.168.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.2.2.0/24</td>
<td>192.168.2.2</td>
<td>100</td>
<td>0</td>
<td>50000</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 172.16.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Multihoming to Receive the Full Internet Routing Table

Perform this task to configure your network to build neighbor relationships with other routers in other autonomous systems while filtering outbound routes. In this task the full Internet routing table will be received from the service providers in the neighboring autonomous systems but only locally originated routes will be advertised to the service providers. This task is configured at Router B in the figure above and uses an access list to permit only locally originated routes and a route map to ensure that only the locally originated routes are advertised outbound to other autonomous systems.

Be aware that receiving the full Internet routing table from two ISPs may use all the memory in smaller routers.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast] vrf vrf-name
5. network network-number [mask network-mask]
6. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
7. neighbor {ip-address|peer-group-name} route-map map-name{in | out}
8. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
9. neighbor {ip-address|peer-group-name} route-map map-name{in | out}
10. exit
11. exit
12. ip as-path access-list access-list-number {deny | permit} as-regular-expression
13. route-map map-name [permit] deny] sequence-number]
14. match as-path path-list-number
15. end
16. show ip bgp [network] [network-mask]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# router bgp 45000</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [unicast</td>
<td>multicast</td>
</tr>
</tbody>
</table>
| **Example:** | - The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.  
  - The **multicast** keyword specifies IPv4 multicast address prefixes.  
  - The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. |
| **Step 5** network network-number [mask network-mask] | Specifies a network as local to this autonomous system and adds it to the BGP routing table.  
- For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates. |
| **Example:** | Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0 |
| **Step 6** neighbor [ip-address | peer-group-name] remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. |
| **Example:** | Router(config-router-af)# neighbor 192.168.1.2 remote-as 40000 |
| **Step 7** neighbor [ip-address | peer-group-name] route-map map-name{in | out} | Applies a route map to incoming or outgoing routes.  
- In this example, the route map named localonly is applied to outbound routes to Router A. |
<p>| <strong>Example:</strong> | Router(config-router-af)# neighbor 192.168.1.2 route-map localonly out |
| <strong>Step 8</strong> neighbor [ip-address | peer-group-name] remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. |
| <strong>Example:</strong> | Router(config-router-af)# neighbor 192.168.3.2 remote-as 50000 |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 9** neighbor \\ {ip-address | peer-group-name} route-map map-name {in | out} | Applies a route map to incoming or outgoing routes.  
- In this example, the route map named localonly is applied to outbound routes to Router E. |
| **Example:** | 
Router(config-router-af)# neighbor 192.168.1.2 route-map localonly out |
| **Step 10** exit | Exits address family configuration mode and enters router configuration mode. |
| **Example:** | 
Router(config-router-af)# exit |
| **Step 11** exit | Exits router configuration mode and enters global configuration mode. |
| **Example:** | 
Router(config-router)# exit |
| **Step 12** ip as-path access-list access-list-number {deny | permit} as-regular-expression | Defines a BGP-related access list.  
- In this example, the access list number 10 is defined to permit only locally originated BGP routes. |
| **Example:** | 
Router(config)# ip as-path access-list 10 permit ^$ |
| **Step 13** route-map map-name [permit | deny] [sequence-number] | Configures a route map and enters route map configuration mode.  
- In this example, a route map named localonly is created. |
| **Example:** | 
Router(config)# route-map localonly permit 10 |
| **Step 14** match as-path path-list-number | Matches a BGP autonomous system path access list.  
- In this example, the BGP autonomous system path access list created in Step 12 is used for the match clause. |
| **Example:** | 
Router(config-route-map)# match as-path 10 |
### Command or Action | Purpose
--- | ---
**Step 15** | end  
Example:  
`Router(config-route-map)# end` | Exits route map configuration mode and enters privileged EXEC mode.

**Step 16** | show ip bgp [network] [network-mask]  
Example:  
`Router# show ip bgp` | Displays the entries in the BGP routing table.  
**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

### Examples

The following example shows the BGP routing table for Router B in the figure above after this task has been configured. Note that the routing table contains the information about the networks in the autonomous systems 40000 and 50000.

```
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*: 10.1.1.0/24      192.168.1.2              0             0 40000 i
*: 10.2.2.0/24      192.168.3.2              0             0 50000 i
*: 172.17.1.0/24    0.0.0.0                  0         32768 i
```

### Configuring BGP Policies

The tasks in this section help you configure BGP policies that filter the traffic in your BGP network. The following optional tasks demonstrate some of the various methods by which traffic can be filtered in your BGP network:

- Filtering BGP Prefixes with Prefix Lists, page 208
- Filtering BGP Prefixes with AS-path Filters, page 212
- Filtering Traffic Using Community Lists, page 215
- Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers, page 219
- Filtering Traffic Using Extended Community Lists, page 222
- Filtering Traffic Using a BGP Route Map Policy List, page 226
- Filtering Traffic Using Continue Clauses in a BGP Route Map, page 230

### Filtering BGP Prefixes with Prefix Lists

Perform this task to use prefix lists to filter BGP route information. The task is configured at Router B in the figure below where both Router A and Router E are set up as BGP peers. A prefix list is configured to permit only routes from the network 10.2.2.0/24 to be outbound. In effect, this will restrict the information
that is received from Router E to be forwarded to Router A. Optional steps are included to display the
prefix list information and to reset the hit count.

**Figure 21** BGP Topology for Configuring BGP Policies Tasks

![BGP Topology](image)

**Note**
The `neighbor prefix-list` and the `neighbor distribute-list` commands are mutually exclusive for a BGP peer.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. network network-number [mask network-mask]
5. neighbor ip-address remote-as autonomous-system-number
6. Repeat Step 5 for all BGP peers.
7. aggregate-address address mask [as-set]
8. neighbor ip-address prefix-list list-name {in | out}
9. exit
10. ip prefix-list list-name [seq seq-number] {deny network / length| permit network / length} [ge ge-value] [le le-value] [eq eq-value]
11. end
12. show ip prefix-list [detail | summary] [prefix-list-name[seq seq-number | network / length [longer | first-match]]]
13. clear ip prefix-list {* | ip-address | peer-group-name} out
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:**  
  Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:**  
  Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| **Example:**  
  Router(config)# router bgp 45000 |
| **Step 4** network network-number [mask network-mask] | (Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.  
  - For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates. |
| **Example:**  
  Router(config-router)# network 172.17.1.0  
  mask 255.255.255.0 |
| **Step 5** neighbor ip-address remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system BGP neighbor table of the local router. |
| **Example:**  
  Router(config-router)# neighbor 192.168.1.2  
  remote-as 40000 |
| **Step 6** Repeat Step 5 for all BGP peers. | -- |
| **Step 7** aggregate-address address mask [as-set] | Creates an aggregate entry in a BGP routing table.  
  - A specified route must exist in the BGP table.  
  - Use the aggregate-address command with no keywords to create an aggregate entry if any more-specific BGP routes are available that fall in the specified range. |
| **Example:**  
  Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 | **Note** Only partial syntax is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference. |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td>`neighbor ip-address prefix-list list-name {in</td>
<td>out}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router)# neighbor 192.168.1.2 prefix-list super172 out</code></td>
<td>In this example, a prefix list called super172 is set for outgoing routes to Router A.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-router)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
</tr>
<tr>
<td>`ip prefix-list list-name [seq seq-number] {deny network / length</td>
<td>permit network / length} [ge ge-value] [le le-value] [eq eq-value]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# ip prefix-list super172 permit 172.0.0.0/8</code></td>
<td>In this example, the prefix list called super172 is defined to permit only route 172.0.0.0/8 to be forwarded. All other routes will be denied because there is an implicit deny at the end of all prefix lists.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code></td>
<td>Exits access list configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-access-list)# end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td></td>
</tr>
<tr>
<td>`show ip prefix-list [detail</td>
<td>summary] [prefix-list-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# show ip prefix-list detail super172</code></td>
<td>In this example, details of the prefix list named super172 will be displayed, including the hit count. Hit count is the number of times the entry has matched a route.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td></td>
</tr>
<tr>
<td>`clear ip prefix-list {*</td>
<td>ip-address</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# clear ip prefix-list super172 out</code></td>
<td>In this example, the hit count for the prefix list called super172 will be reset.</td>
</tr>
</tbody>
</table>
Examples

The following output from the `show ip prefix-list` command shows details of the prefix list named `super172`, including the hit count. The `clear ip prefix-list` command is entered to reset the hit count and the `show ip prefix-list` command is entered again to show the hit count reset to 0.

```
Router# show ip prefix-list detail super172
ip prefix-list super172:
    count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
    seq 5 permit 172.0.0.0/8 (hit count: 1, refcount: 1)
Router# clear ip prefix-list super172
Router# show ip prefix-list detail super172
ip prefix-list super172:
    count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
    seq 5 permit 172.0.0.0/8 (hit count: 0, refcount: 1)
```

Filtering BGP Prefixes with AS-path Filters

Perform this task to filter BGP prefixes using AS-path filters with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list is configured at Router B in the figure above. The first line of the access list denies all matches to the AS-path 50000 and the second line allows all other paths. The router uses the `neighbor filter-list` command to specify the AS-path access list as an outbound filter. After the filtering is enabled, traffic can be received from both Router A and Router E but updates originating from autonomous system 50000 (Router E) are not forwarded by Router B to Router A. If any updates from Router E originated from another autonomous system, they would be forwarded because they would contain both autonomous system 50000 plus another autonomous system number, and that would not match the AS-path access list.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. network network-number [mask network-mask]
5. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
6. Repeat Step 5 for all BGP peers.
7. neighbor {ip-address | peer-group-name} filter-list {access-list-number} [in | out]
8. exit
9. ip as-path access-list access-list-number {deny | permit} as-regular-expression
10. Repeat Step 9 for all entries required in the AS-path access list.
11. end
12. show ip bgp regexp as-regular-expression
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
| Example:          |         |
| Router> enable    |         |
| **Step 2** configure terminal | Enters global configuration mode.  
| Example:          |         |
| Router# configure terminal |         |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| Example:          |         |
| Router(config)# router bgp 45000 |         |
| **Step 4** network network-number [mask network-mask] | (Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.  
| Example:          |         |
| Router(config-router)# network 172.17.1.0 mask 255.255.255.0 |         |
| **Step 5** neighbor {ip-address | peer-group-name} remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router. |
| Example:          |         |
| Router(config-router)# neighbor 192.168.1.2 remote-as 40000 |         |
| **Step 6** Repeat Step 5 for all BGP peers. | -- |
| **Step 7** neighbor {ip-address | peer-group-name} filter-list{access-list-number}{in | out} | Distributes BGP neighbor information as specified in a prefix list.  
<p>| Example:          |         |
| Router(config-router)# neighbor 192.168.1.2 filter-list 100 out |         |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> ip as-path access-list access-list-number [deny</td>
<td>permit] as-regular-expression</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip as-path access-list 100 deny ^50000$</td>
<td>• In the first example, access list number 100 is defined to deny any AS-path that starts and ends with 50000.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip as-path access-list 100 permit .*</td>
<td>• In the second example, all routes that do not match the criteria in the first example of the AS-path access list will be permitted. The period and asterisk symbols imply that all characters in the AS-path will match so Router B will forward those updates to Router A.</td>
</tr>
<tr>
<td><strong>Step 10</strong> Repeat Step 9 for all entries required in the AS-path access list.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 11</strong> end</td>
<td>Exits access list configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-access-list)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> show ip bgp regexp as-regular-expression</td>
<td>Displays routes matching the regular expression.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp regexp ^50000$</td>
<td>• To verify the regular expression you can use this command.</td>
</tr>
<tr>
<td></td>
<td>• In this example, all paths that match the expression &quot;starts and ends with 50000&quot; will be displayed.</td>
</tr>
</tbody>
</table>

**Examples**

The following output from the **show ip bgp regexp** command shows the autonomous system paths that match the regular expression—start and end with AS-path 50000:

```plaintext
Router# show ip bgp regexp ^50000$
BGP table version is 9, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, v valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```
**Filtering Traffic Using Community Lists**

Perform this task to filter traffic by creating BGP community lists and then reference them within a route map to control incoming routes. BGP communities provide a method of filtering inbound or outbound routes for large, complex networks. Instead of compiling long access or prefix lists of individual peers, BGP allows grouping of peers with identical routing policies even though they reside in different autonomous systems or networks.

In this task, Router B in the figure above is configured with several route maps and community lists to control incoming routes.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
5. neighbor {ip-address | peer-group-name} route-map route-map-name [in | out]
6. exit
7. route-map map-name [permit | deny] [sequence-number]
8. match community {standard-list-number | expanded-list-number | community-list-name [exact]}
9. set weight weight
10. exit
11. route-map map-name [permit | deny] [sequence-number]
12. match community {standard-list-number | expanded-list-number | community-list-name [exact]}
13. set community community-number
14. exit
15. ip community-list {standard-list-number | standard list-name {deny | permit} [community-number] [AA:NN] [internet] [local-AS] [no-advertise] [no-export]} | {expanded-list-number | expanded list-name {deny | permit} regular-expression}
16. Repeat Step 15 to create all the required community lists.
17. exit
18. show ip community-list [standard-list-number | expanded-list-number | community-list-name] [exact-match]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.3.2 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor {ip-address</td>
<td>peer-group-name} route-map route-map-name [in</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.3.2 route-map 2000 in</td>
<td>• In this example, the route map called 2000 is applied to inbound routes from the BGP peer at 192.168.3.2.</td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> route-map map-name [permit</td>
<td>deny] [sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# route-map 2000 permit 10</td>
<td>• In this example, the route map called 2000 is defined.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 8</strong> match community {standard-list-number</td>
<td>expanded-list-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# match community 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> set weight weight</td>
<td>Specifies the BGP weight for the routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# set weight 30</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> route-map map-name [permit</td>
<td>deny] [sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# route-map 3000 permit 10</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> match community {standard-list-number</td>
<td>expanded-list-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# match community 2</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> set community community-number</td>
<td>Sets the BGP communities attribute.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# set community 99</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> exit</td>
<td>Exits route map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-route-map)# exit</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 15</strong> ip community-list [standard-list-number] standard list-name [deny</td>
<td>permit] [community-number] [AA:NN] [internet] [local-AS] [no-advertise] [no-export]]</td>
</tr>
<tr>
<td></td>
<td>• In the first example, community list 1 permits routes with a community attribute of 100. Router C routes all have community attribute of 100 so their weight will be set to 30.</td>
</tr>
<tr>
<td></td>
<td>• In the second example, community list 2 effectively permits all routes by using the <strong>internet</strong> keyword. Any routes that did not match community list 1 are checked against community list 2. All routes are permitted but no changes are made to the route attributes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong> Two examples are shown here because the task example requires both these statements to be configured.</td>
</tr>
<tr>
<td>Router(config)# ip community-list 1 permit 100</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip community-list 2 permit internet</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> Repeat Step 15 to create all the required community lists.</td>
<td><strong>--</strong></td>
</tr>
<tr>
<td><strong>Step 17</strong> exit</td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> show ip community-list [standard-list-number] expanded-list-number [community-list-name] [exact-match]</td>
<td>Displays configured BGP community list entries.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip community-list 1</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

The following sample output verifies that community list 1 has been created, with the output showing that community list 1 permits routes with a community attribute of 100:

```
Router# show ip community-list 1
Community standard list 1
    permit 100
```

The following sample output verifies that community list 2 has been created, with the output showing that community list 2 effectively permits all routes by using the **internet** keyword:

```
Router# show ip community-list 2
```
Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers

In Cisco IOS Release 2.4 and later releases, BGP support for 4-octet (4-byte) autonomous system numbers was introduced. The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format, for example, Router B is in autonomous system number 65538 in the figure below. For more details about the introduction of 4-byte autonomous system numbers, see BGP Autonomous System Number Formats, page 167.

Perform this task to filter BGP prefixes with AS-path filters using 4-byte autonomous system numbers with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list is configured at Router B in the figure below. The first line of the access list denies all matches to the AS-path 65550 and the second line allows all other paths. The router uses the `neighbor filter-list` command to specify the AS-path access list as an outbound filter. After the filtering is enabled, traffic can be received from both Router A and Router E but updates originating from autonomous system 65550 (Router E) are not forwarded by Router B to Router A. If any updates from Router E originated from another autonomous system, they would be forwarded because they would contain both autonomous system 65550 plus another autonomous system number, and that would not match the AS-path access list.

---

**Note**

In Cisco IOS XE Release 2.1 and later releases, the maximum number of autonomous system access lists that can be configured with the `ip as-path access-list` command is increased from 199 to 500.

---

**Figure 22** BGP Topology for Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast] vrf vrf-name
5. network network-number [mask network-mask]
6. neighbor {ip-address | peer-group-name} remote-as autonomous-system-number
7. Repeat Step 6 for all BGP peers.
8. neighbor {ip-address | peer-group-name} filter-list access-list-number{in | out}
9. exit
10. ip as-path access-list access-list-number {deny | permit} as-regular-expression
11. Repeat Step 10 for all entries required in the AS-path access list.
12. end
13. show ip bgp regexp as-regular-expression

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# router bgp 65538</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>address-family ipv4 [unicast</td>
<td>multicast] vrf vrf-name]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td>Specifies the IPv4 address family and enters address family configuration mode.</td>
</tr>
<tr>
<td>- The <strong>unicast</strong> keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the <strong>unicast</strong> keyword is not specified with the <strong>address-family ipv4</strong> command.</td>
<td></td>
</tr>
<tr>
<td>- The <strong>multicast</strong> keyword specifies IPv4 multicast address prefixes.</td>
<td></td>
</tr>
<tr>
<td>- The <strong>vrf</strong> keyword and <strong>vrf-name</strong> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>network network-number [mask network-mask]</td>
<td>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</td>
<td>- For exterior protocols the <strong>network</strong> command controls which networks are advertised. Interior protocols use the <strong>network</strong> command to determine where to send updates.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only partial syntax is used in this example. For more details, see the <em>Cisco IOS IP Routing: BGP Command Reference</em>.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.1.2 remote-as 65536</td>
<td>- In this example, the IP address for the neighbor at Router A is added.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>Repeat Step 6 for all BGP peers.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td>neighbor {ip-address</td>
<td>peer-group-name} filter-list access-list-number {in</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.1.2 filter-list 99 out</td>
<td>- In this example, an access list number 99 is set for outgoing routes to Router A.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td>Exits router configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 10
ip as-path access-list access-list-number
{deny | permit} as-regular-expression

#### Example:

```bash
Router(config)# ip as-path access-list 99 deny ^65550$
```

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines a BGP-related access list and enters access list configuration mode.</td>
</tr>
<tr>
<td>- In the first example, access list number 99 is defined to deny any AS-path that starts and ends with 65550.</td>
</tr>
<tr>
<td>- In the second example, all routes that do not match the criteria in the first example of the AS-path access list will be permitted. The period and asterisk symbols imply that all characters in the AS-path will match, so Router B will forward those updates to Router A.</td>
</tr>
</tbody>
</table>

#### Note

Two examples are shown here because the task example requires both these statements to be configured.

#### Step 11
Repeat Step 10 for all entries required in the AS-path access list.

#### Step 12
end

#### Example:

```bash
Router(config-access-list)# end
```

#### Step 13
show ip bgp regexp as-regular-expression

#### Example:

```bash
Router# show ip bgp regexp ^65550$
```

### Examples

The following output from the `show ip bgp regexp` command shows the autonomous system paths that match the regular expression—start and end with AS-path 65550:

```bash
Router# show ip bgp regexp ^65550$
BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
*> 10.2.2.0/24 192.168.3.2 0 0 65550 i
```

### Filtering Traffic Using Extended Community Lists

Perform this task to filter traffic by creating an extended BGP community list to control outbound routes. BGP communities provide a method of filtering inbound or outbound routes for large, complex networks.
Instead of compiling long access or prefix lists of individual peers, BGP allows grouping of peers with identical routing policies even though they reside in different autonomous systems or networks.

In this task, Router B in the figure above is configured with an extended named community list to specify that the BGP peer at 192.168.1.2 is not sent advertisements about any path through or from autonomous system 50000. The IP extended community-list configuration mode is used and the ability to resequence entries is shown.

A sequence number is applied to all extended community list entries by default regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode and not in global configuration mode.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** `{expanded-list-number | expanded list-name | standard-list-number | standard list-name}`
4. `[sequence-number] {deny[regular-expression] | exit | permit[regular-expression]}
5. Repeat Step 4 for all the required permit or deny entries in the extended community list.
6. **resequence** `[starting-sequence][sequence-increment]
7. **exit**
8. **router bgp** autonomous-system-number
9. **network** network-number `[mask network-mask]
10. **neighbor** `{ip-address | peer-group-name} remote-as` autonomous-system-number
11. Repeat Step 10 for all the required BGP peers.
12. **end**
13. **show ip extcommunity-list** `[list-number | list-name]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router&gt; enable</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router# configure terminal</strong></td>
</tr>
</tbody>
</table>
### Filtering Traffic Using Extended Community Lists

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters IP extended community-list configuration mode to create or configure an extended community list.</td>
</tr>
<tr>
<td>`ip extcommunity-list {expanded-list-number</td>
<td>expanded list-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip extcommunity-list expanded DENY50000</td>
<td></td>
</tr>
</tbody>
</table>

**Step 4**  
| `[sequence-number] {deny[regular-expression] | exit | permit[regular-expression]}` | Configures an expanded community list entry. |
|-------------------|---------|
| • In the first example, an expanded community list entry with the sequence number 10 is configured to deny advertisements about paths from autonomous system 50000. |
| • In the second example, an expanded community list entry with the sequence number 20 is configured to deny advertisements about paths through autonomous system 50000. |
| **Example:** | |
| Router(config-extcomm-list)# 10 deny _50000_ | |
| **Example:** | |
| and | |
| Router(config-extcomm-list)# 20 deny ^50000 .* | |
| **Note** | Two examples are shown here because the task example requires both these statements to be configured. |
| **Note** | Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference. |

**Step 5**  
| Repeat Step 4 for all the required permit or deny entries in the extended community list. | -- |

**Step 6**  
<table>
<thead>
<tr>
<th><code>resequence [starting-sequence][sequence-increment]</code></th>
<th>Resequences expanded community list entries.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• In this example, the sequence number of the first expanded community list entry is set to 50 and subsequent entries are set to increment by 100. The second expanded community list entry is therefore set to 150.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-extcomm-list)# resequence 50 100</td>
<td></td>
</tr>
</tbody>
</table>

**Step 7**  
<table>
<thead>
<tr>
<th><code>exit</code></th>
<th>Exits expanded community-list configuration mode and enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-extcomm-list)# exit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 8</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config)# router bgp 45000
```

<table>
<thead>
<tr>
<th><strong>Step 9</strong> network network-number [mask network-mask]</th>
<th>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</th>
</tr>
</thead>
</table>

**Example:**

```
Router(config-router)# network 172.17.1.0 mask 255.255.255.0
```

- **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.

| **Step 10** neighbor {ip-address | peer-group-name} remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router. |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|

**Example:**

```
Router(config-router)# neighbor 192.168.3.2 remote-as 50000
```

<table>
<thead>
<tr>
<th><strong>Step 11</strong> Repeat Step 10 for all the required BGP peers.</th>
<th>--</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Step 12</strong> end</th>
<th>Exits router configuration mode and enters privileged EXEC mode.</th>
</tr>
</thead>
</table>

**Example:**

```
Router(config-router)# end
```

| **Step 13** show ip extcommunity-list [list-number | list-name] | Displays configured BGP expanded community list entries. |
|----------------------------------------------------------|----------------------------------------------------------|

**Example:**

```
Router# show ip extcommunity-list
DENY50000
```

**Examples**

The following sample output verifies that the BGP expanded community list DENY50000 has been created, with the output showing that the entries to deny advertisements about autonomous system 50000 have been resequenced from 10 and 20 to 50 and 150:

```
Router# show ip extcommunity-list 1
Expanded extended community-list DENY50000
```
Filtering Traffic Using a BGP Route Map Policy List

Perform this task to create a BGP policy list and then reference it within a route map.

A policy list is like a route map that contains only match clauses. With policy lists there are no changes to match clause semantics and route map functions. The match clauses are configured in policy lists with permit and deny statements and the route map evaluates and processes each match clause to permit or deny routes based on the configuration. AND and OR semantics in the route map function the same way for policy lists as they do for match clauses.

Policy lists simplify the configuration of BGP routing policy in medium-size and large networks. The network operator can reference preconfigured policy lists with groups of match clauses in route maps and easily apply general changes to BGP routing policy. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.

Perform this task to create a BGP policy list to filter traffic that matches the autonomous system path and MED of a router and then create a route map to reference the policy list.

BGP routing must be configured in your network and BGP neighbors must be established.

Note

- BGP route map policy lists do not support the configuration of IP version 6 (IPv6) match clauses in policy lists.
- Policy lists support only match clauses and do not support set clauses. However, policy lists can coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists.
- Policy lists are supported only by BGP. They are not supported by other IP routing protocols. This limitation does not interfere with normal operations of a route map, including redistribution, because policy list functions operate transparently within BGP and are not visible to other IP routing protocols.
- Policy lists support only match clauses and do not support set clauses. However, policy lists can coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists. The first route map example configures AND semantics, and the second route map configuration example configures semantics. Both examples in this section show sample route map configurations that reference policy lists and separate match and set clauses in the same configuration.
SUMMARY STEPS

1. enable
2. configure terminal
3. ip policy-list policy-list-name {permit | deny}
4. match as-path as-number
5. match metric metric
6. exit
7. route-map map-name [permit | deny] [sequence-number]
8. match ip address {access-list-number | access-list-name} [... access-list-number | ... access-list-name]
9. match policy-list policy-list-name
10. set community community-number [additive] [well-known-community] | none
11. set local-preference preference-value
12. end
13. show ip policy-list [policy-list-name]
14. show route-map [route-map-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3 ip policy-list policy-list-name {permit</td>
<td>deny}</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# ip policy-list POLICY_LIST_NAME-1 permit</td>
</tr>
<tr>
<td>Step 4 match as-path as-number</td>
<td>Creates a match clause to permit routes from the specified autonomous system path.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-policy-list)# match as-path 40000</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 5</strong> match metric <em>metric</em></td>
<td>Creates a match clause to permit routes with the specified metric.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-policy-list)# match metric 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits policy list configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-policy-list)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> route-map *map-name [permit</td>
<td>deny] [sequence-number]*</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# route-map MAP-NAME-1 permit 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> match ip address {access-list-number</td>
<td>access-list-name} [... access-list-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# match ip address 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> match policy-list <em>policy-list-name</em></td>
<td>Creates a clause that will match the specified policy list.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# match policy-list POLICY-LIST-NAME-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> set community *community-number [additive] [well-known-community]</td>
<td>Creates a clause to set or remove the specified community.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-route-map)# set community 10: 1</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 11</th>
<th><code>set local-preference</code> preference-value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| ```
Router(config-route-map)#
set local-preference 140
``` |
| **Purpose:** | Creates a clause to set the specified local preference value. |

<table>
<thead>
<tr>
<th>Step 12</th>
<th><code>end</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| ```
Router(config-route-map)# end
``` |
| **Purpose:** | Exits route map configuration mode and enters privileged EXEC mode. |

<table>
<thead>
<tr>
<th>Step 13</th>
<th><code>show ip policy-list [policy-list-name]</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| ```
Router# show ip policy-list POLICY-LIST-NAME-1
``` |
| **Purpose:** | Displays information about configured policy lists and policy list entries. |

<table>
<thead>
<tr>
<th>Step 14</th>
<th><code>show route-map [route-map-name]</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| ```
Router# show route-map
``` |
| **Purpose:** | Displays locally configured route maps and route map entries. |

### Examples

The following sample output verifies that a policy list has been created, with the output displaying the policy list name and configured match clauses:

```
Router# show ip policy-list
POLICY-LIST-NAME-1
policy-list POLICY-LIST-NAME-1 permit
    Match clauses:
        metric 20
        as-path {as-path filter}: 1
```

### Note

A policy list name can be specified when the `show ip policy-list` command is entered. This option can be useful for filtering the output of this command and verifying a single policy list.

### Examples

The following sample output from the `show route-map` command verifies that a route map has been created and a policy list is referenced. The output of this command displays the route map name and policy lists that are referenced by the configured route maps.

```
Router# show route-map
route-map ROUTE-MAP-NAME-1, deny, sequence 10
    Match clauses:
        Set clauses:
            Policy routing matches: 0 packets, 0 bytes
    route-map ROUTE-MAP-NAME-1, permit, sequence 10
```
Filtering Traffic Using Continue Clauses in a BGP Route Map

Perform this task to filter traffic using continue clauses in a BGP route map.

**Note**

Continue clauses can go only to a higher route map entry (a route map entry with a higher sequence number) and cannot go to a lower route map entry.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
5. neighbor {ip-address|peer-group-name} route-map map-name{in | out}
6. exit
7. route-map map-name {permit | deny} [sequence-number]
8. match ip address {access-list-number | access-list-name} [... access-list-number | ... access-list-name]
9. set community community-number [additive] [well-known-community] | none
10. continue [sequence-number]
11. end
12. show route-map [map-name]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables higher privilege levels, such as privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode, and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 10.0.0.1 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor {ip-address</td>
<td>peer-group-name} route-map map-name{in</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> route-map map-name {permit</td>
<td>deny} [sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# route-map ROUTE-MAP-NAME permit 10</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 8</strong> match ip address [access-list-number</td>
<td>access-list-name] [...] access-list-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Configures a match command that specifies the conditions under which policy routing and route filtering occur.</td>
</tr>
<tr>
<td>Router(config-route-map)# match ip address 1</td>
<td>- Multiple match commands can be configured. If a match command is configured, a match must occur in order for the continue statement to be executed. If a match command is not configured, set and continue clauses will be executed.</td>
</tr>
<tr>
<td><strong>Step 9</strong> set community community-number [additive] [well-known-community]</td>
<td>Configures a set command that specifies the routing action to perform if the criteria enforced by the match commands are met.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>- Multiple set commands can be configured.</td>
</tr>
<tr>
<td>Router(config-route-map)# set community 10: 1</td>
<td>- In this example, a clause is created to set the specified community.</td>
</tr>
<tr>
<td><strong>Step 10</strong> continue [sequence-number]</td>
<td>Configures a route map to continue to evaluate and execute match statements after a successful match occurs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>- If a sequence number is configured, the continue clause will go to the route map with the specified sequence number.</td>
</tr>
<tr>
<td>Router(config-route-map)# continue</td>
<td>- If no sequence number is specified, the continue clause will go to the route map with the next sequence number. This behavior is called an &quot;implied continue.&quot;</td>
</tr>
<tr>
<td><strong>Step 11</strong> end</td>
<td><strong>Note</strong> Continue clauses in outbound route maps are supported in Cisco IOS XE Release 2.1 and later releases.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Exits route-map configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Router(config-route-map)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> show route-map [map-name]</td>
<td>(Optional) Displays locally configured route maps. The name of the route map can be specified in the syntax of this command to filter the output.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show route-map</td>
</tr>
</tbody>
</table>
Examples

The following sample output shows how to verify the configuration of continue clauses using the `show route-map` command. The output displays configured route maps including the match, set, and continue clauses.

```
Router# show route-map
route-map MARKETING, permit, sequence 10
Match clauses:
  ip address (access-lists): 1
  metric 10
Continue: sequence 40
Set clauses:
  as-path prepend 10
Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
Match clauses:
  ip address (access-lists): 2
  metric 20
Set clauses:
  as-path prepend 10 10
Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
Match clauses:
  to next entry 40
Set clauses:
  as-path prepend 10 10 10
Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 40
Match clauses:
  community (community-list filter): 10:1
Set clauses:
  local-preference 104
Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
Match clauses:
Set clauses:
  community 655370
Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for Connecting to a Service Provider Using External BGP

- Influencing Inbound Path Selection Examples, page 234
- Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example, page 234
- Influencing Outbound Path Selection Examples, page 236
- Filtering BGP Prefixes with Prefix Lists Examples, page 237
- Filtering Traffic Using Community Lists Examples, page 238
- Filtering Traffic Using AS-path Filters Example, page 239
- Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers Examples, page 240
- Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example, page 240
- Filtering Traffic Using a BGP Route Map Example, page 243
- Filtering Traffic Using Continue Clauses in a BGP Route Map Example, page 243
Influencing Inbound Path Selection Examples

The following example shows how you can use route maps to modify incoming data from a neighbor. Any route received from 10.222.1.1 that matches the filter parameters set in autonomous system access list 200 will have its weight set to 200 and its local preference set to 250, and it will be accepted.

```
router bgp 100
  neighbor 10.222.1.1 route-map FIX-WEIGHT in
  neighbor 10.222.1.1 remote-as 1
  ip as-path access-list 200 permit ^690$
  ip as-path access-list 200 permit ^1800
  route-map FIX-WEIGHT permit 10
    match as-path 200
    set local-preference 250
    set weight 200
```

In the following example, the route map named finance marks all paths originating from autonomous system 690 with an MED metric attribute of 127. The second permit clause is required so that routes not matching autonomous system path list 1 will still be sent to neighbor 10.1.1.1.

```
router bgp 65000
  neighbor 10.1.1.1 route-map finance out
  ip as-path access-list 1 permit ^690_
  ip as-path access-list 2 permit .*
  route-map finance permit 10
    match as-path 1
    set metric 127
  route-map finance permit 20
    match as-path 2
```

Inbound route maps could perform prefix-based matching and set various parameters of the update. Inbound prefix matching is available in addition to autonomous system path and community list matching. The following example shows how the `set local-preference` route map configuration command sets the local preference of the inbound prefix 172.20.0.0/16 to 120:

```
! router bgp 65100
  network 10.108.0.0
  neighbor 10.108.1.1 remote-as 65200
  neighbor 10.108.1.1 route-map set-local-pref in
  ! route-map set-local-pref permit 10
    match ip address 2
    set local preference 120
  ! route-map set-local-pref permit 20
    access-list 2 permit 172.20.0.0 0.0.255.255
    access-list 2 deny any
```

Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example

This example shows how to configure BGP to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. In Cisco IOS XE Release 2.4 and later releases, BGP support for 4-octet (4-byte) autonomous system numbers was introduced. The 4-byte autonomous
system numbers in this example are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure below. For more details about the introduction of 4-byte autonomous system numbers, see BGP Autonomous System Number Formats, page 167.

One of the methods that BGP can use to influence the choice of paths in another autonomous system is to modify the AS-path attribute. For example, in the figure below, Router A advertises its own network, 172.17.1.0, to its BGP peers in autonomous system 65538 and autonomous system 65550. When the routing information is propagated to autonomous system 65545, the routers in autonomous system 65545 have network reachability information about network 172.17.1.0 from two different routes. The first route is from autonomous system 65538 with an AS-path consisting of 65538, 65536. The second route is through autonomous system 65547 with an AS-path of 65547, 65550, 65536. If all other BGP attribute values are the same, Router C in autonomous system 65545 would choose the route through autonomous system 65538 for traffic destined for network 172.17.1.0 because it is the shortest route in terms of autonomous systems traversed.

Autonomous system 65536 now receives all traffic from autonomous system 65545 for the 172.17.1.0 network through Router B in autonomous system 65538. If, however, the link between autonomous system 65538 and autonomous system 65536 is a really slow and congested link, the set as-path prepend command can be used at Router A to influence inbound path selection for the 172.17.1.0 network by making the route through autonomous system 65538 appear to be longer than the path through autonomous system 65550. The configuration is done at Router A in the figure below by applying a route map to the outbound BGP updates to Router B. Using the set as-path prepend command, all the outbound BGP updates from Router A to Router B will have their AS-path attribute modified to add the local autonomous system number 65536 twice. After the configuration, autonomous system 65545 receives updates about the 172.17.1.0 network through autonomous system 65538. The new AS-path is 65538, 65536, 65536, 65536, which is now longer than the AS-path from autonomous system 65547 (unchanged at a value of 65547, 65550, 65536). Networking devices in autonomous system 65545 will now prefer the route through autonomous system 65547 to forward packets with a destination address in the 172.17.1.0 network.

**Figure 23** Network Topology for Modifying the AS-path Attribute
The configuration for this example is performed at Router A in the figure above.

```
router bgp 65536
  address-family ipv4 unicast
    network 172.17.1.0 mask 255.255.255.0
    neighbor 192.168.1.2 remote-as 65538
    neighbor 192.168.1.2 activate
    neighbor 192.168.1.2 route-map PREPEND out
  exit-address-family
  exit
  route-map PREPEND permit 10
  set as-path prepend 65536 65536
```

Influencing Outbound Path Selection Examples

The following example creates an outbound route filter and configures Router-A (10.1.1.1) to advertise the filter to Router-B (172.16.1.2). An IP prefix list named FILTER is created to specify the 192.168.1.0/24 subnet for outbound route filtering. The ORF send capability is configured on Router-A so that Router-A can advertise the outbound route filter to Router-B.

**Router-A Configuration (Sender)**

```
ip prefix-list FILTER seq 10 permit 192.168.1.0/24
!
router bgp 65100
  address-family ipv4 unicast
  neighbor 172.16.1.2 remote-as 65200
  neighbor 172.16.1.2 ebgp-multihop
  neighbor 172.16.1.2 capability orf prefix-list send
  neighbor 172.16.1.2 prefix-list FILTER in
end
```

**Router-B Configuration (Receiver)**

The following example configures Router-B to advertise the ORF receive capability to Router-A. Router-B will install the outbound route filter, defined in the FILTER prefix list, after ORF capabilities have been exchanged. An inbound soft reset is initiated on Router-B at the end of this configuration to activate the outbound route filter.

```
router bgp 65200
  address-family ipv4 unicast
  neighbor 10.1.1.1 remote-as 65100
  neighbor 10.1.1.1 ebgp-multihop 255
  neighbor 10.1.1.1 capability orf prefix-list receive
end

clear ip bgp 10.1.1.1 in prefix-filter
```

The following example shows how the route map named set-as-path is applied to outbound updates to the neighbor 10.69.232.70. The route map will prepend the autonomous system path "65100 65100" to routes that pass access list 1. The second part of the route map is to permit the advertisement of other routes.

```
router bgp 65100
  network 172.16.0.0
  network 172.17.0.0
  neighbor 10.69.232.70 remote-as 65200
  neighbor 10.69.232.70 route-map set-as-path out
  !
  route-map set-as-path 10 permit
    match address 1
    set as-path prepend 65100 65100
    !
  route-map set-as-path 20 permit
```
match address 2
!
access-list 1 permit 172.16.0.0 0.0.255.255
access-list 1 permit 172.17.0.0 0.0.255.255
!
access-list 2 permit 0.0.0.0 255.255.255.255

Filtering BGP Prefixes with Prefix Lists Examples

This section contains the following examples:

- Filtering BGP Prefixes Using a Single Prefix List, page 237
- Filtering BGP Prefixes Using a Group of Prefixes, page 238
- Adding or Deleting Prefix List Entries, page 238

Filtering BGP Prefixes Using a Single Prefix List

The following example shows how a prefix list denies the default route 0.0.0.0/0:

```
ip prefix-list abc deny 0.0.0.0/0
```

The following example shows how a prefix list permits a route that matches the prefix 10.0.0.0/8:

```
ip prefix-list abc permit 10.0.0.0/8
```

The following example shows how to configure the BGP process so that it accepts only prefixes with a prefix length of /8 to /24:

```
router bgp 40000
  network 10.20.20.0
distribute-list prefix max24 in
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
```

The following example configuration shows how to conditionally originate a default route (0.0.0.0/0) in RIP when a prefix 10.1.1.0/24 exists in the routing table:

```
ip prefix-list cond permit 10.1.1.0/24
!
routemapt default-condition permit 10
  match ip address prefix-list cond
!
routerr rip
default-information originate route-map default-condition
```

The following example shows how to configure BGP to accept routing updates from 192.168.1.1 only, besides filtering on the prefix length:

```
routerr bgp 40000
distribute-list prefix max24 gateway allowlist in
!
ip prefix-list allowlist seq 5 permit 192.168.1.1/32
```

The following example shows how to direct the BGP process to filter incoming updates to the prefix using name1, and match the gateway (next hop) of the prefix being updated to the prefix list name2, on GigabitEthernet interface 0/0/0:

```
routerr bgp 103
distribute-list prefix name1 gateway name2 in gigabitethernet 0/0/0
```
Filtering BGP Prefixes Using a Group of Prefixes

The following example shows how to configure BGP to permit routes with a prefix length up to 24 in network 192/8:

```
ip prefix-list abc permit 192.0.0.0/8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in 192/8:

```
ip prefix-list abc deny 192.0.0.0/8 ge 25
```

The following example shows how to configure BGP to permit routes with a prefix length greater than 8 and less than 24 in all address space:

```
ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in all address space:

```
ip prefix-list abc deny 0.0.0.0/0 ge 25
```

The following example shows how to configure BGP to deny all routes in network 10/8, because any route in the Class A network 10.0.0.0/8 is denied if its mask is less than or equal to 32 bits:

```
ip prefix-list abc deny 10.0.0.0/8 le 32
```

The following example shows how to configure BGP to deny routes with a mask greater than 25 in 192.168.1.0/24:

```
ip prefix-list abc deny 192.168.1.0/24 ge 25
```

The following example shows how to configure BGP to permit all routes:

```
ip prefix-list abc permit 0.0.0.0/0 le 32
```

Adding or Deleting Prefix List Entries

You can add or delete individual entries in a prefix list if a prefix list has the following initial configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 192.168.0.0/15
```

The following example shows how to delete an entry from the prefix list so that 192.168.0.0 is not permitted, and add a new entry that permits 10.0.0.0/8:

```
no ip prefix-list abc permit 192.168.0.0/15
ip prefix-list abc permit 10.0.0.0/8
```

The new configuration is as follows:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 10.0.0.0/8
```

Filtering Traffic Using Community Lists Examples

This section contains two examples of the use of BGP communities with route maps.
The first example shows how the route map named set-community is applied to the outbound updates to the neighbor 172.16.232.50. The routes that pass access list 1 have the special community attribute value no-export. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```conf
router bgp 100
    neighbor 172.16.232.50 remote-as 200
    neighbor 172.16.232.50 send-community
    neighbor 172.16.232.50 route-map set-community out

route-map set-community permit 10
    match address 1
    set community no-export

route-map set-community permit 20
    match address 2
```

The second example shows how the route map named set-community is applied to the outbound updates to neighbor 172.16.232.90. All the routes that originate from autonomous system 70 have the community values 200 200 added to their already existing values. All other routes are advertised as normal.

```conf
route-map bgp 200
    neighbor 172.16.232.90 remote-as 100
    neighbor 172.16.232.90 send-community
    neighbor 172.16.232.90 route-map set-community out

route-map set-community permit 10
    match as-path 1
    set community 200 200 additive

route-map set-community permit 20

ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
```

Filtering Traffic Using AS-path Filters Example

The following example shows BGP path filtering by neighbor. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.12.10. Similarly, only routes passing access list 3 will be accepted from 192.168.12.10.

```conf
router bgp 200
    neighbor 192.168.12.10 remote-as 100
    neighbor 192.168.12.10 filter-list 1 out
    neighbor 192.168.12.10 filter-list 2 in
exit
ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit _100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```
Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers Examples

Asplain Default Format in Cisco IOS XE Release 2.4 and Later Releases
The following example is available in Cisco IOS XE Release 2.4 and later releases, and shows BGP path filtering by neighbor using 4-byte autonomous system numbers in asplain format. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.3.2.

```bash
ip as-path access-list 2 permit ^65536$
router bgp 65538
address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 65550
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 filter-list 2 in
end
```

Asdot Default Format in Cisco IOS XE Release 2.3
The following example available in Cisco IOS XE Release 2.3 shows BGP path filtering by neighbor using 4-byte autonomous system numbers in asdot format. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.3.2.

```bash
ip as-path access-list 2 permit ^1\.0$
router bgp 1.2
address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.3.2 filter-list 2 in
end
```

Note
In Cisco IOS XE Release 2.4 and later releases, this example works if you have configured asdot as the default display format using the `bgp asnotation dot` command.

Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example

Asplain Default Format in Cisco IOS XE Release 2.4 and Later Releases
The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS XE Release 2.4 and later releases, extended BGP communities support 4-byte autonomous system numbers in the regular expressions in asplain by default. Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The `ip extcommunity-list` command is used to configure named or numbered extended community lists. All of the
standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

**Figure 24** BGP Topology for Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers in Asplain Format

A sequence number is applied to all extended community list entries by default regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode and not in global configuration mode.

The figure above is configured with an extended named community list to specify that the BGP peer at 192.168.1.2 is not sent advertisements about any path through or from the 4-byte autonomous system 65550. The IP extended community-list configuration mode is used, and the ability to resequence entries is shown.

```plaintext
ip extcommunity-list expanded DENY65550
10 deny _65550_
20 deny ^65550 .*
resequence 50 100
exit
router bgp 65538
network 172.17.1.0 mask 255.255.255.0
address-family ipv4 unicast
   neighbor 192.168.3.2 remote-as 65550
   neighbor 192.168.1.2 remote-as 65536
   neighbor 192.168.3.2 activate
   neighbor 192.168.1.2 activate
end
show ip extcommunity-list DENY65550
```

**Asdot Default Format in Cisco IOS XE Release 2.3**

The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS XE Release 2.3, extended BGP communities support 4-byte autonomous system numbers in the regular expressions in asdot format only. Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The `ip extcommunity-list` command is used to configure named or numbered extended community lists. All of the standard rules
of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

**Note**

In Cisco IOS XE Release 2.4 and later releases, this example works if you have configured asdot as the default display format using the `bgp asnotation dot` command.

**Note**

A sequence number is applied to all extended community list entries by default regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode and not in global configuration mode.

The figure above is configured with an extended named community list to specify that the BGP peer at 192.168.1.2 is not sent advertisements about any path through or from the 4-byte autonomous system 65550. The IP extended community-list configuration mode is used, and the ability to resequence entries is shown.

```plaintext
ip extcommunity-list expanded DENY114
10 deny _1\.14_
20 deny ^1\.14 .*
resequence 50 100
exit
router bgp 1.2
network 172.17.1.0 mask 255.255.255.0
address-family ipv4 unicast
neighbor 192.168.3.2 remote-as 1.14
neighbor 192.168.1.2 remote-as 1.0
neighbor 192.168.3.2 activate
neighbor 192.168.1.2 activate
end
show ip extcommunity-list DENY114
```
Filtering Traffic Using a BGP Route Map Example

The following example shows how to use an address family to configure BGP so that any unicast and multicast routes from neighbor 10.1.1.1 are accepted if they match access list 1:

```plaintext
route-map filter-some-multicast
  match ip address 1
  exit
router bgp 65538
  neighbor 10.1.1.1 remote-as 65537
  address-family ipv4 unicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
  exit
exit
router bgp 65538
  neighbor 10.1.1.1 remote-as 65537
  address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
end
```

Filtering Traffic Using Continue Clauses in a BGP Route Map Example

The following example shows continue clause configuration in a route map sequence.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful matches occurs. If a match does not occur, the route map will “fall through” to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries. Only the first successful match ip address clause is supported.

If a successful match does not occur in route map entry 20, the route map will “fall through” to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed and the continue clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will “fall through” to route map entry 30 and execute the set clause. A sequence number is not specified for the continue clause so route map entry 40 will be evaluated.

There are two behaviors that can occur when the same `set` command is repeated in subsequent continue clause entries. For `set` commands that configure an additive or accumulative value (for example, `set community additive`, `set extended community additive`, and `set as-path prepend`), subsequent values are added by subsequent entries. The following example illustrates this behavior. After each set of match clauses, a `set as-path prepend` command is configured to add an autonomous system number to the as-path. After a match occurs, the route map stops evaluating match clauses and starts executing the set clauses, in the order in which they were configured. Depending on how many successful match clauses occur, the as-path is prepended by one, two, or three autonomous system numbers.

```plaintext
route-map ROUTE-MAP-NAME permit 10
  match ip address 1
  match metric 10
  set as-path prepend 10
  continue 30
!
route-map ROUTE-MAP-NAME permit 20
  match ip address 2
  match metric 20
  set as-path prepend 10 10
!
route-map ROUTE-MAP-NAME permit 30
```
set as-path prepend 10 10 10
continue
!
route-map ROUTE-MAP-NAME permit 40
match community 10:1
set local-preference 104

In this example, the same set command is repeated in subsequent continue clause entries but the behavior is different from the first example. For set commands that configure an absolute value, the value from the last instance will overwrite the previous value(s). The following example illustrates this behavior. The set clause value in sequence 20 overwrites the set clause value from sequence 10. The next hop for prefixes from the 172.16/16 network is set to 10.2.2.2 and not 10.1.1.1.

ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
match ip address prefix-list 1
set ip next hop 10.1.1.1
continue 20
exit
route-map RED permit 20
match ip address prefix-list 2
set ip next hop 10.2.2.2
end

---

**Note**

Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. The following example illustrates this case.

In the following example, when routes match an as-path of 10, 20, or 30, the routes are permitted and the continue clause jumps over the explicit deny clause to process the match ip address prefix list. If a match occurs here, the route metric is set to 100. Only routes that do not match an as-path of 10, 20, or 30 and do match a community number of 30 are denied. To deny other routes, you must configure an explicit deny statement.

route-map test permit 10
match as-path 10 20 30
continue 30
exit
route-map test deny 20
match community 30
exit
route-map test permit 30
match ip address prefix-list 1
set metric 100
exit

---

**Where to Go Next**

- To configure advanced BGP feature tasks, proceed to the "Configuring Advanced BGP Features" module.
- To configure BGP neighbor session options, proceed to the "Configuring BGP Neighbor Session Options" module.
- To configure internal BGP tasks, proceed to the "Configuring Internal BGP Features" module.
Additional References

The following sections provide references related to connecting to a service provider using external BGP.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>BGP overview</td>
<td>&quot;Cisco BGP Overview&quot; module</td>
</tr>
<tr>
<td>Configuring basic BGP tasks</td>
<td>&quot;Configuring a Basic BGP Network&quot; module</td>
</tr>
<tr>
<td>BGP fundamentals and description</td>
<td>Large-Scale IP Network Solutions, Khalid Raza and Mark Turner, Cisco Press, 2000</td>
</tr>
<tr>
<td>Implementing and controlling BGP in scalable networks</td>
<td>Building Scalable Cisco Networks, Catherine Paquet and Diane Teare, Cisco Press, 2001</td>
</tr>
<tr>
<td>Interdomain routing basics</td>
<td>Internet Routing Architectures, Bassam Halabi, Cisco Press, 1997</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT SAFI</td>
<td>MDT SAFI</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-BGP4-MIB</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1772</td>
<td>Application of the Border Gateway Protocol in the Internet</td>
</tr>
<tr>
<td>RFC 1773</td>
<td>Experience with the BGP Protocol</td>
</tr>
</tbody>
</table>
Connecting to a Service Provider Using External BGP

Feature Information for Connecting to a Service Provider Using External BGP

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.

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1774</td>
<td>BGP-4 Protocol Analysis</td>
</tr>
<tr>
<td>RFC 1930</td>
<td>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</td>
</tr>
<tr>
<td>RFC 2519</td>
<td>A Framework for Inter-Domain Route Aggregation</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
<tr>
<td>RFC 3392</td>
<td>Capabilities Advertisement with BGP-4</td>
</tr>
<tr>
<td>RFC 4271</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 4893</td>
<td>BGP Support for Four-Octet AS Number Space</td>
</tr>
<tr>
<td>RFC 5396</td>
<td>Textual Representation of Autonomous system (AS) Numbers</td>
</tr>
<tr>
<td>RFC 5398</td>
<td>Autonomous System (AS) Number Reservation for Documentation Use</td>
</tr>
</tbody>
</table>

Technical Assistance.

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>
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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

Table 17  Feature Information for Connecting to a Service Provider Using External BGP

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| BGP Increased Support of Numbered AS-Path Access Lists to 500 | Cisco IOS XE Release 2.1       | The BGP Increased Support of Numbered AS-Path Access Lists to 500 feature increases the maximum number of autonomous systems access lists that can be configured using the `ip as-path access-list` command from 199 to 500.  
This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. |
| BGP Named Community Lists                         | Cisco IOS XE Release 2.1       | The BGP Named Community Lists feature introduces a new type of community list called the named community list. The BGP Named Community Lists feature allows the network operator to assign meaningful names to community lists and increases the number of community lists that can be configured. A named community list can be configured with regular expressions and with numbered community lists. All rules of numbered communities apply to named community lists except that there is no limitation on the number of community attributes that can be configured for a named community list.  
This feature was introduced on the Cisco ASR 1000 Series Routers. |

---
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Prefix-Based Outbound Route Filtering</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Prefix-Based Outbound Route Filtering feature uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers. Configuring this feature can help reduce the amount of system resources required for generating and processing routing updates by filtering out unwanted routing updates at the source. For example, this feature can be used to reduce the amount of processing required on a router that is not accepting full routes from a service provider network. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP 4 Prefix Filter and In-bound Route Maps</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP Route-Map Continue</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Route-Map Continue feature introduces the continue clause to BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>
### Feature Information for Connecting to a Service Provider Using External BGP

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Route-Map Continue Support for an Outbound Policy</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP Route-Map Policy List Support</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Route-Map Policy List Support feature introduces new functionality to BGP route maps. This feature adds the capability for a network operator to group route map match clauses into named lists called policy lists. A policy list functions like a macro. When a policy list is referenced in a route map, all of the match clauses are evaluated and processed as if they had been configured directly in the route map. This enhancement simplifies the configuration of BGP routing policy in medium-size and large networks because a network operator can preconfigure policy lists with groups of match clauses and then reference these policy lists within different route maps. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| BGP Support for 4-Byte ASN | Cisco IOS XE Release 2.3 Cisco IOS XE Release 2.4 | The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295. In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte autonomous system numbers uses asdot as the only configuration format, regular expression match, and output display, with no asplain support. In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the `bgp asnotation dot` command. The following commands were introduced or modified by this feature: `bgp asnotation dot, bgp confederation identifier, bgp confederation peers, all clear ip bgp commands that configure an autonomous system number, ip as-path access-list, ip extcommunity-list, match source-protocol, neighbor local-as, neighbor remote-as, redistribute (IP), router bgp, route-target, set as-path, set...`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Support for Named Extended Community Lists</td>
<td>Cisco IOS XE Release 2.1</td>
<td><em>extcommunity</em>, <em>set origin</em>, all <em>show ip bgp</em> commands that display an autonomous system number, and <em>show ip extcommunity-list</em>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The BGP Support for Named Extended Community Lists feature introduces the ability to configure extended community lists using names in addition to the existing numbered format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP Support for Sequenced Entries in Extended Community Lists</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Support for Sequenced Entries in Extended Community Lists feature introduces automatic sequencing of individual entries in BGP extended community lists. This feature also introduces the ability to remove or resequence extended community list entries without deleting the entire existing extended community list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring BGP Neighbor Session Options

This module describes configuration tasks to configure various options involving Border Gateway Protocol (BGP) neighbor peer sessions. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. This module contains tasks that use BGP neighbor session commands to configure:

- Fast session deactivation
- Bidirectional Forwarding Detection (BFD) for BGP IPv6 neighbors
- A router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down
- Options to help an autonomous system migration
- TTL Security Check, a lightweight security mechanism to protect External BGP (eBGP) peering sessions from CPU-utilization-based attacks
- BGP dynamic neighbors

- Finding Feature Information, page 253
- Prerequisites for Configuring BGP Neighbor Session Options, page 253
- Restrictions for Configuring BGP Neighbor Session Options, page 254
- Information About Configuring BGP Neighbor Session Options, page 254
- How to Configure BGP Neighbor Session Options, page 259
- Configuration Examples for BGP Neighbor Session Options, page 292
- Where to Go Next, page 298
- Additional References, page 298
- Feature Information for Configuring BGP Neighbor Session Options, page 300

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Configuring BGP Neighbor Session Options

Before configuring advanced BGP features you should be familiar with the "Cisco BGP Overview" module and the "Configuring a Basic BGP Network" module.
Restrictions for Configuring BGP Neighbor Session Options

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.

Information About Configuring BGP Neighbor Session Options

- BGP Neighbor Sessions, page 254
- BGP Support for Fast Peering Session Deactivation, page 254
- BFD Support of BGP IPv6 Neighbors, page 255
- BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached, page 255
- BGP Network Autonomous System Migration, page 256
- TTL Security Check for BGP Neighbor Sessions, page 257
- BGP Support for TCP Path MTU Discovery per Session, page 258
- BGP Dynamic Neighbors, page 259

BGP Neighbor Sessions

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. A BGP-speaking router does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking routers.

A BGP neighbor device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a peer instead of neighbor because a neighbor may imply the idea that the BGP devices are directly connected with no other router in between. Configuring BGP neighbor or peer sessions uses BGP neighbor session commands so this module will prefer the use of the term "neighbor" over "peer."

BGP Support for Fast Peering Session Deactivation

- BGP Hold Timer, page 254
- BGP Fast Peering Session Deactivation, page 254
- Selective Address Tracking for BGP Fast Session Deactivation, page 255

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco IOS XE software. This timer value is set as the default to protect the BGP routing process from instability that can be caused by peering sessions with other routing protocols. BGP routers typically carry large routing tables, so frequent session resets are not desirable.

BGP Fast Peering Session Deactivation
BGP fast peering session deactivation improves BGP convergence and response time to adjacency changes with BGP neighbors. This feature is event driven and configured on a per-neighbor basis. When this feature is enabled, BGP will monitor the peering session with the specified neighbor. Adjacency changes are detected and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

Selective Address Tracking for BGP Fast Session Deactivation

In Cisco IOS XE Release 2.1 and later releases, the BGP Selective Address Tracking feature introduced the use of a route map with BGP fast session deactivation. The `route-map` keyword and `map-name` argument are used with the `neighbor fall-over` BGP neighbor session command to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset. The route map is not used for session establishment.

Note

Only `match ip address` and `match source-protocol` commands are supported in the route map. No `set` commands or other `match` commands are supported.

BFD Support of BGP IPv6 Neighbors

In Cisco IOS XE Release 3.3S and later releases, Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that have an IPv6 address. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. BFD provides faster reconvergence time for BGP after a forwarding path failure.

BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached

- Prefix Limits and BGP Peering Sessions, page 255
- BGP Neighbor Session Restart with the Maximum Prefix Limit, page 255

Prefix Limits and BGP Peering Sessions

There is a configurable limit on the maximum number of prefixes that a router that is running BGP can receive from a peer router. This limit is configured with the `neighbor maximum-prefix` command. When the router receives too many prefixes from a peer router and the maximum-prefix limit is exceeded, the peering session is disabled or brought down. The session stays down until the network operator manually brings the session back up by entering the `clear ip bgp` command. Entering the `clear ip bgp` command clears stored prefixes.

BGP Neighbor Session Restart with the Maximum Prefix Limit

In Cisco IOS XE Release 2.1 and later releases, the restart keyword was added to enhance the capabilities of the neighbor maximum-prefix command. This enhancement allows the network operator to configure a router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down. There is a configurable time interval at which peering can be reestablished automatically. The configurable timer argument for the restart keyword is specified in minutes. The time range is from 1 to 65,535 minutes.
BGP Network Autonomous System Migration

- Autonomous System Migration for BGP Networks, page 256
- Dual Autonomous System Support for BGP Network Autonomous System Migration, page 256

**Autonomous System Migration for BGP Networks**

Autonomous-system migration can be necessary when a telecommunications or Internet service provider purchases another network. It is desirable for the provider to be able to integrate the second autonomous system without disrupting existing customer peering arrangements. The amount of configuration required in the customer networks can make this a cumbersome task that is difficult to complete without disrupting service.

**Dual Autonomous System Support for BGP Network Autonomous System Migration**

In Cisco IOS XE Release 2.1 and later releases, support was added for dual BGP autonomous system configuration to allow a secondary autonomous system to merge under a primary autonomous system, without disrupting customer peering sessions. The configuration of this feature is transparent to customer networks. Dual BGP autonomous system configuration allows a router to appear, to external peers, as a member of secondary autonomous system during the autonomous system migration. This feature allows the network operator to merge the autonomous systems and then later migrate customers to new configurations during normal service windows without disrupting existing peering arrangements.

The neighbor local-as command is used to customize the AS_PATH attribute by adding and removing autonomous system numbers for routes received from eBGP neighbors. This feature allows a router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. This feature simplifies this process of changing the autonomous system number in a BGP network by allowing the network operator to merge a secondary autonomous system into a primary autonomous system and then later update the customer configurations during normal service windows without disrupting existing peering arrangements.

**BGP Autonomous System Migration Support for Confederations, Individual Peering Sessions, and Peer Groupings**

This feature supports confederations, individual peering sessions, and configurations applied through peer groups and peer templates. If this feature is applied to a group peers, the individual peers cannot be customized.

**Ingress Filtering During BGP Autonomous System Migration**

Autonomous system path customization increases the possibility that routing loops can be created if such customization is misconfigured. The larger the number of customer peerings, the greater the risk. You can minimize this possibility by applying policies on the ingress interfaces to block the autonomous system number that is in transition or routes that have no local-as configuration.
BGP prepends the autonomous system number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This feature should be configured only for autonomous system migration and should be deconfigured after the transition has been completed. This procedure should be attempted only by an experienced network operator, as routing loops can be created with improper configuration.

---

TTL Security Check for BGP Neighbor Sessions

- BGP Support for the TTL Security Check, page 257
- TTL Security Check for BGP Neighbor Sessions, page 257
- TTL Security Check Support for Multihop BGP Neighbor Sessions, page 258
- Benefits of the BGP Support for TTL Security Check, page 258

BGP Support for the TTL Security Check

When implemented for BGP, the TTL Security Check feature introduces a lightweight security mechanism to protect eBGP neighbor sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses.

The TTL Security Check feature protects the eBGP neighbor session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP neighbor session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is unnecessary.

Although it is possible to forge the TTL field in an IP packet header, accurately forging the TTL count to match the TTL count from a trusted peer is impossible unless the network to which the trusted peer belongs has been compromised.

The TTL Security Check feature supports both directly connected neighbor sessions and multihop eBGP neighbor sessions. The BGP neighbor session is not affected by incoming packets that contain invalid TTL values. The BGP neighbor session will remain open, and the router will silently discard the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

TTL Security Check for BGP Neighbor Sessions

The BGP Support for TTL Security Check feature is configured with the `neighbor ttl-security` command in router configuration mode or address family configuration mode. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. The `hop-count` argument is used to configure the maximum number of hops that separate the two peers. The TTL value is determined by the router from the configured hop count. The value for this argument is a number from 1 to 254.
TTL Security Check Support for Multihop BGP Neighbor Sessions

The BGP Support for TTL Security Check feature supports both directly connected neighbor sessions and multihop neighbor sessions. When this feature is configured for a multihop neighbor session, the `neighbor ebgp-multihop` router configuration command cannot be configured and is not needed to establish the neighbor session. These commands are mutually exclusive, and only one command is required to establish a multihop neighbor session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.

To configure this feature for an existing multihop session, you must first disable the existing neighbor session with the `no neighbor ebgp-multihop` command. The multihop neighbor session will be restored when you enable this feature with the `neighbor ttl-security` command.

This feature should be configured on each participating router. To maximize the effectiveness of this feature, the `hop-count` argument should be strictly configured to match the number of hops between the local and external network. However, you should also consider path variation when configuring this feature for a multihop neighbor session.

Benefits of the BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect eBGP neighbor sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP autonomous system.

BGP Support for TCP Path MTU Discovery per Session

- Path MTU Discovery, page 258
- BGP Neighbor Session TCP PMTUD, page 259

Path MTU Discovery

The IP protocol family was designed to use a wide variety of transmission links. The maximum IP packet length is 65000 bytes. Most transmission links enforce a smaller maximum packet length limit, called the maximum transmission unit (MTU), which varies with the type of the transmission link. The design of IP accommodates link packet length limits by allowing intermediate routers to fragment IP packets as necessary for their outgoing links. The final destination of an IP packet is responsible for reassembling its fragments as necessary.

All TCP sessions are bounded by a limit on the number of bytes that can be transported in a single packet, and this limit is known as the maximum segment size (MSS). TCP breaks up packets into chunks in a transmit queue before passing packets down to the IP layer. A smaller MSS may not be fragmented at an IP device along the path to the destination device, but smaller packets increase the amount of bandwidth needed to transport the packets. The maximum TCP packet length is determined by both the MTU of the outbound interface on the source device and the MSS announced by the destination device during the TCP setup process.

Path MTU discovery (PMTUD) was developed as a solution to the problem of finding the optimal TCP packet length. PMTUD is an optimization (detailed in RFC 1191) wherein a TCP connection attempts to send the longest packets that will not be fragmented along the path from source to destination. It does this by using a flag, don’t fragment (DF), in the IP packet. This flag is supposed to alter the behavior of an
intermediate router that cannot send the packet across a link because it is too long. Normally the flag is off, and the router should fragment the packet and send the fragments. If a router tries to forward an IP datagram, with the DF bit set, to a link that has a lower MTU than the size of the packet, the router will drop the packet and return an ICMP Destination Unreachable message to the source of this IP datagram, with the code indicating “fragmentation needed and DF set.” When the source device receives the ICMP message, it will lower the send MSS, and when TCP retransmits the segment, it will use the smaller segment size.

**BGP Neighbor Session TCP PMTUD**

TCP path MTU discovery is enabled by default for all BGP neighbor sessions, but there are situations when you may want to disable TCP path MTU discovery for one or all BGP neighbor sessions. Although PMTUD works well for larger transmission links (for example, Packet over Sonet links), a badly configured TCP implementation or a firewall may slow or stop the TCP connections from forwarding any packets. In this type of situation, you may need to disable TCP path MTU discovery. In Cisco IOS XE Release 2.1 and later releases, configuration options were introduced to permit TCP path MTU discovery to be disabled, or subsequently reenabled, either for a single BGP neighbor session or for all BGP sessions. To disable the TCP path MTU discovery globally for all BGP neighbors, use the `no bgp transport path-mtu-discovery` command in router configuration mode. To disable the TCP path MTU discovery for a single neighbor, use the `no neighbor transport path-mtu-discovery` command in router or address family configuration modes. For more details, see the Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 276 or the Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 279.

**BGP Dynamic Neighbors**

Support for the BGP Dynamic Neighbors feature was introduced in Cisco IOS XE Release 3.1S. BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups.

After a subnet range is configured for a BGP peer group and a TCP session is initiated by another router for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. After the initial configuration of subnet ranges and activation of the peer group (referred to as a `listen range group`), dynamic BGP neighbor creation does not require any further CLI configuration on the initial router. Other routers can establish a BGP session with the initial router, but the initial router need not establish a BGP session to other routers if the IP address of the remote peer used for the BGP session is not within the configured range.

To support the BGP Dynamic Neighbors feature, the output for the `show ip bgp neighbors`, `show ip bgp peer-group`, and `show ip bgp summary` commands was updated to display information about dynamic neighbors.

A dynamic BGP neighbor will inherit any configuration for the peer group. In larger BGP networks, implementing BGP dynamic neighbors can reduce the amount and complexity of CLI configuration and save CPU and memory usage. Only IPv4 peering is supported.

**How to Configure BGP Neighbor Session Options**

- Configuring Fast Session Deactivation, page 260
- Configuring BFD for BGP IPv6 Neighbors, page 264
Configuring Fast Session Deactivation

The tasks in this section show how to configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about route dampening, see the "Configuring Internal BGP Features" module.

- Configuring Fast Session Deactivation for a BGP Neighbor, page 260
- Configuring Selective Address Tracking for Fast Session Deactivation, page 261

Configuring Fast Session Deactivation for a BGP Neighbor

Perform this task to establish a peering session with a BGP neighbor and then configure the peering session for fast session deactivation to improve the network convergence time if the peering session is deactivated.

Enabling fast session deactivation for a BGP neighbor can significantly improve BGP convergence time. However, unstable IGP peers can still introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name]`
5. `neighbor ip-address remote-as autonomous-system-number`
6. `neighbor ip-address fall-over`
7. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

Example:

```
Router> enable
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [mdt</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Establishes a peering session with a BGP neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.1 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address fall-over</td>
<td>Configures the BGP peering to use fast session deactivation.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.1 fall-over</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring Selective Address Tracking for Fast Session Deactivation**

Perform this task to configure selective address tracking for fast session deactivation. The optional `route-map` keyword and `map-name` argument of the `neighbor fall-over` command are used to determine if a peering session with a BGP neighbor should be deactivated (reset) when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset.
Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. `router bgp autonomous-system-number`
4. `neighbor {ip-address|peer-group-name} remote-as autonomous-system-number`
5. `neighbor ip-address fall-over [route-map map-name]`
6. exit
7. `ip prefix-list list-name [seq seq-value] {deny network I length | permit network I length} [ge ge-value] [le le-value]`
8. `route-map map-name [permit | deny][sequence-number]`
9. `match ip address prefix-list prefix-list-name [prefix-list-name...]`
10. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 <code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example: Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring BGP Neighbor Session Options**

**Configuring Selective Address Tracking for Fast Session Deactivation**

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> neighbor [ip-address</td>
<td>peer-group-name] remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 192.168.1.2 remote-as 40000</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 5** neighbor ip-address fall-over [route-map map-name] | Applies a route map when a route to the BGP changes.  
• In this example, the route map named CHECK-NBR is applied when the route to neighbor 192.168.1.2 changes. |
| **Example:** Router(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR |  |
| **Step 6** exit | Exits router configuration mode and enters global configuration mode. |
| **Example:** Router(config-router)# exit |  |
| **Step 7** ip prefix-list list-name [seq seq-value]|(deny  
network / length | permit network / length)] [ge ge-value] [le le-value] | Creates a prefix list for BGP next-hop route filtering.  
• Selective next-hop route filtering supports prefix length matching or source protocol matching on a per-address-family basis.  
• The example creates a prefix list named FILTER28 that permits routes only if the mask length is greater than or equal to 28. |
| **Example:** Router(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28 |  |
| **Step 8** route-map map-name [permit | deny][sequence-number] | Configures a route map and enters route-map configuration mode.  
• In this example, a route map named CHECK-NBR is created. If there is an IP address match in the following match command, the IP address will be permitted. |
| **Example:** Router(config)# route-map CHECK-NBR permit 10 |  |
| **Step 9** match ip address prefix-list prefix-list-name [prefix-list-name...] | Matches the IP addresses in the specified prefix list.  
• Use the prefix-list-name argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. |
| **Example:** Router(config-route-map)# match ip address prefix-list FILTER28 |  |
| **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference. |
### Command or Action | Purpose
---|---
Step 10 end | Exits configuration mode and returns to privileged EXEC mode.

**Example:**

```
Router(config-route-map)# end
```

- [What to Do Next, page 264](#)

**What to Do Next**

The BGP Support for Next-Hop Address Tracking feature improves the response time of BGP to next-hop changes for routes installed in the RIB, which can also improve overall BGP convergence. For information about BGP next-hop address tracking, see the "Configuring Advanced BGP Features" module.

**Configuring BFD for BGP IPv6 Neighbors**

In Cisco IOS XE Release 3.3S and later releases, Bidirectional Forwarding Detection (BFD) can be used for BGP neighbors that have an IPv6 address.

Once it has been verified that BFD neighbors are up, the `show bgp ipv6 unicast neighbors` command will indicate that BFD is being used to detect fast fallover on the specified neighbor.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ipv6 unicast-routing
4. ipv6 cef
5. interface type number
6. ipv6 address ipv6-address / prefix-length
7. bfd interval milliseconds min_rx milliseconds multiplier multiplier-value
8. no shutdown
9. exit
10. router bgp autonomous-system-number
11. no bgp default ipv4-unicast
12. address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6]
13. neighbor ipv6-address remote-as autonomous-system-number
14. neighbor ipv6-address fall-over bfd
15. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ipv6 unicast-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv6 cef</td>
<td>Enables Cisco Express Forwarding for IPv6.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ipv6 cef</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type number</td>
<td>Configures an interface type and number.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface fastethernet 0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ipv6 address ipv6-address / prefix-length</td>
<td>Configures an IPv6 address and enables IPv6 processing on an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ipv6 address 2001:DB8:1:1::1/64</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> bfd interval milliseconds min_rx milliseconds multiplier multiplier-value</td>
<td>Sets the baseline BFD session parameters on an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# bfd interval 500 min_rx 500 multiplier 3</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 8</strong> no shutdown</td>
<td>Restarts an interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# no shutdown</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 9** exit | Exits interface configuration mode and enters global configuration mode. |
| **Example:** | |
| Router(config-if)# exit | |

| **Step 10** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| **Example:** | |
| Router(config)# router bgp 40000 | |

| **Step 11** no bgp default ipv4-unicast | Disables the default IPv4 unicast address family for establishing peering sessions. |
| **Example:** | |
| Router(config-router)# no bgp default ipv4-unicast | |
| • We recommend configuring this command in the global scope. | |

| **Step 12** address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6] | Enters address family configuration mode and enables IPv6 addressing. |
| **Example:** | |
| Router(config-router)# address-family ipv6 | |

| **Step 13** neighbor ipv6-address remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table of the local router. |
| **Example:** | |
| Router(config-router-af)# neighbor 2001:DB8:2:1::4 remote-as 45000 | |

| **Step 14** neighbor ipv6-address fall-over bfd | Enables BGP to monitor the peering session of an IPv6 neighbor using BFD. |
| **Example:** | |
| Router(config-router)# neighbor 2001:DB8:2:1::4 fall-over bfd | |
Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded

Perform this task to configure the time interval at which a BGP neighbor session is reestablished by a router when the number of prefixes that have been received from a BGP peer has exceeded the maximum prefix limit.

The network operator can configure a router that is running BGP to automatically reestablish a neighbor session that has been brought down because the configured maximum-prefix limit has been exceeded. No intervention from the network operator is required when this feature is enabled.

**Note**

This task attempts to reestablish a disabled BGP neighbor session at the configured time interval that is specified by the network operator. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. The network operator will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the `warning-only` keyword of the `neighbor maximum-prefix` command can be configured to disable the restart capability, while the network operator corrects the underlying problem.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor {ip-address | peer-group-name} maximum-prefix maximum [threshold] [restart minutes] [warning-only]
5. end
6. show ip bgp neighbors ip-address
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| Example: | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example: | Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode and creates a BGP routing process. |
| Example: | Router(config)# router bgp 101 |
| **Step 4** neighbor [ip-address | peer-group-name] maximum-prefix [maximum] [threshold] [restart minutes] [warning-only] | Configures the maximum-prefix limit on a router that is running BGP.  
- Use the **restart** keyword and **minutes** argument to configure the router to automatically reestablish a neighbor session that has been disabled because the maximum-prefix limit has been exceeded. The configurable range of **minutes** is from 1 to 65535 minutes.  
- Use the **warning-only** keyword to configure the router to disable the restart capability to allow you to fix a peer that is sending too many prefixes.  
**Note** If the **minutes** argument is not configured, the disabled session will stay down after the maximum-prefix limit is exceeded. This is the default behavior. |
| Example: | Router(config-router)# neighbor 10.4.9.5 maximum-prefix 1000 90 restart 60 |
| **Step 5** end | Exits configuration mode and enters privilaged EXEC mode. |
| Example: | Router(config-router)# end |
| **Step 6** show ip bgp neighbors ip-address | (Optional) Displays information about the TCP and BGP connections to neighbors.  
- In this example, the output from this command will display the maximum prefix limit for the specified neighbor and the configured restart timer value.  
**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*. |
Examples

The following example output from the `show ip bgp neighbors` command verifies that a router has been configured to automatically reestablish disabled neighbor sessions. The output shows that the maximum prefix limit for neighbor 10.4.9.5 is set to 1000 prefixes, the restart threshold is set to 90 percent, and the restart interval is set at 60 minutes.

```
Router# show ip bgp neighbors 10.4.9.5

BGP neighbor is 10.4.9.5, remote AS 101, internal link
BGP version 4, remote router ID 10.4.9.5
Last read 00:00:14, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Address family IPv4 Unicast: advertised and received
Message statistics:
  InQ depth is 0
  OutQ depth is 0
  Sent Open: 1, Rcvd: 1
  Sent Notifications: 0, Rcvd: 0
  Sent Updates: 0, Rcvd: 0
  Sent Keepalives: 23095, Rcvd: 23095
  Sent Route Refresh: 0, Rcvd: 0
  Total: 23096, Rcvd: 23096
Default minimum time between advertisement runs is 5 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor versions 1/0 1/0
Output queue sizes : 0 self, 0 replicated
Index 2, Offset 0, Mask Ox4
Member of update-group 2

Prefix activity:

<table>
<thead>
<tr>
<th>Prefix activity</th>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefixes Current:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prefixes Total:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Implicit Withdraw:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Explicit Withdraw:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Used as bestpath:</td>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>Used as multipath:</td>
<td>n/a</td>
<td>0</td>
</tr>
</tbody>
</table>

Local Policy Denied Prefixes: ---

Total:

---

!Configured maximum number of prefixes and restart interval information!
Maximum prefixes allowed 1000
Threshold for warning message 90%, restart interval 60 min
Number of NLRI s in the update sent: max 0, min 0
Connections established 1; dropped 0
Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.4.9.21, Local port: 179
Foreign host: 10.4.9.5, Foreign port: 11871
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x5296BD2C):

<table>
<thead>
<tr>
<th>Event Timers</th>
<th>Starts</th>
<th>Wakes</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrans</td>
<td>23098</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>TimeWait</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>AckHold</td>
<td>23096</td>
<td>22692</td>
<td>0x0</td>
</tr>
<tr>
<td>SendWnd</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>KeepAlive</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>GiveUp</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>PmtuAger</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>DeadWait</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
</tbody>
</table>

Iss: 1900546793 sndna: 1900985663 sndnx: 1900985663 sndwnd: 14959
Irs: 2894590641 rcvnxt: 2895029492 rcvwnd: 14978 delrcvwnd: 1406
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRtt: 0 ms, maxRtt: 316 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
Datagrams (max data segment is 1460 bytes):

Rcvd: 46021 (out of order: 0), with data: 23096, total data bytes: 438850
Sent: 46095 (retransmit: 0, fastretransmit: 0), with data: 23097, total data by9
Troubleshooting Tips

Use the `clear ip bgp` command to reset a BGP connection using BGP soft reconfiguration. This command can be used to clear stored prefixes to prevent a router that is running BGP from exceeding the maximum-prefix limit. For more details about using BGP soft reconfiguration, see the "Monitoring and Maintaining Basic BGP" task in the "Configuring a Basic BGP Network" module.

Display of the following error messages can indicate an underlying problem that is causing the neighbor session to become disabled. The network operator should check the values that are configured for the maximum-prefix limit and the configuration of any peers that are sending an excessive number of prefixes. The following sample error messages are similar to the error messages that may be displayed:

```
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Up
00:01:14:%BGP-4-MAXPFX:No. of unicast prefix received from 10.10.10.2 reaches 5, max 6
00:01:14:%BGP-3-MAXPFXEXCEED:No.of unicast prefix received from 10.10.10.2:7 exceed limit6
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Down - BGP Notification sent
00:01:14:%BGP-3-NOTIFICATION: sent to neighbor 10.10.10.2 3/1 (update malformed) 0 byte
```

The `bgp dampening` command can be used to configure the dampening of a flapping route or interface when a peer is sending too many prefixes and causing network instability. Use this command only when troubleshooting or tuning a router that is sending an excessive number of prefixes. For more details about BGP route dampening, see the "Configuring Advanced BGP Features" module.

Configuring Dual-AS Peering for Network Migration

Perform this task to configure a BGP peer router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. When the BGP peer is configured with dual autonomous system numbers then the network operator can merge a secondary autonomous system into a primary autonomous system and update the customer configuration during a future service window without disrupting existing peering arrangements.

The `show ip bgp` and `show ip bgp neighbors` commands can be used to verify autonomous system number for entries in the routing table and the status of this feature.

---

**Note**

- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for only true eBGP peering sessions. This feature cannot be configured for two peers in different subautonomous systems of a confederation.
- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for individual peering sessions and configurations applied through peer groups and peer templates. If this command is applied to a peer group, the peers cannot be individually customized.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. neighbor ip-address local-as [autonomous-system-number [no-prepend [replace-as [dual-as]]]]
6. neighbor ip-address remove-private-as
7. end
8. show ip bgp [network] [network-mask] [longer-prefixes] [prefix-list prefix-list-name] [route-map route-map-name] [shorter-prefixes mask-length]
9. show ip bgp neighbors [neighbor-address] [received-routes | routes | advertised-routes | paths regexp | dampened-routes | received prefix-filter]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3 router bgp autonomous-system-number</td>
<td>Enters router configuration mode, and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# router bgp 40000</td>
</tr>
<tr>
<td>Step 4 neighbor ip-address remote-as autonomous-system-number</td>
<td>Establishes a peering session with a BGP neighbor.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-router)# neighbor 10.0.0.1</td>
</tr>
<tr>
<td></td>
<td>remote-as 45000</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor ip-address local-as [autonomous-system-number [no-prepend [replace-as [dual-as]]]]</td>
<td>Customizes the AS_PATH attribute for routes received from an eBGP neighbor.</td>
</tr>
<tr>
<td></td>
<td>• The replace-as keyword is used to prepend only the local autonomous system number (as configured with the ip-address argument) to the AS_PATH attribute. The autonomous system number from the local BGP routing process is not prepended.</td>
</tr>
<tr>
<td></td>
<td>• The dual-as keyword is used to configure the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous system number configured with the ip-address argument (local-as).</td>
</tr>
<tr>
<td></td>
<td>• The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the local-as number.</td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address remove-private-as</td>
<td>(Optional) Removes private autonomous system numbers from outbound routing updates.</td>
</tr>
<tr>
<td></td>
<td>• This command can be used with the replace-as functionality to remove the private autonomous system number and replace it with an external autonomous system number.</td>
</tr>
<tr>
<td></td>
<td>• Private autonomous system numbers (64512 to 65535) are automatically removed from the AS_PATH attribute when this command is configured.</td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show ip bgp [network] [network-mask] [longer-prefixes] [prefix-list prefix-list-name]</td>
<td>Displays entries in the BGP routing table.</td>
</tr>
<tr>
<td></td>
<td>[route-map route-map-name] [shorter-prefixes mask-length]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Command or Action | Purpose
--- | ---
Step 9 `show ip bgp neighbors [neighbor-address] [received-routes | routes] advertised-routes | paths regexp | dampened-routes | received prefix-filter]` | Displays information about TCP and BGP connections to neighbors.

---

**Example:**

```
Router# show ip bgp neighbors
```

**Configuring the TTL Security Check for BGP Neighbor Sessions**

Perform this task to allow BGP to establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the BGP neighbor session.

- To maximize the effectiveness of the BGP Support for TTL Security Check feature, we recommend that you configure it on each participating router. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router.

**Note**

- The `neighbor ebgp-multihop` command is not needed when the BGP Support for TTL Security Check feature is configured for a multihop neighbor session and should be disabled before configuring this feature.
- The effectiveness of the BGP Support for TTL Security Check feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected neighbor sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of the local and remote network. This restriction also includes peers that are on the network segment between the local and remote network.

**SUMMARY STEPS**

1. `enable`
2. `trace [protocol] destination`
3. `configure terminal`
4. `router bgp autonomous-system-number`
5. `neighbor ip-address ttl-security hops hop-count`
6. `end`
7. `show running-config`
8. `show ip bgp neighbors [ip-address]`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  ```
  Example:
  Router> enable
  ```  
  • Enter your password if prompted. |
| **Step 2** trace [protocol] destination | Discovers the routes of the specified protocol that packets will actually take when traveling to their destination.  
  ```
  Example:
  Router# trace ip 10.1.1.1
  ```  
  • Enter the trace command to determine the number of hops to the specified peer. |
| **Step 3** configure terminal | Enters global configuration mode.  
  ```
  Example:
  Router# configure terminal
  ``` |
| **Step 4** router bgp autonomous-system-number | Enters router configuration mode, and creates a BGP routing process.  
  ```
  Example:
  Router(config)# router bgp 65000
  ``` |
| **Step 5** neighbor ip-address ttl-security hops hop-count | Configures the maximum number of hops that separate two peers.  
  ```
  Example:
  Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
  ```  
  • The hop-count argument is set to the number of hops that separate the local and remote peer. If the expected TTL value in the IP packet header is 254, then the number 1 should be configured for the hop-count argument. The range of values is a number from 1 to 254.  
  • When the BGP Support for TTL Security Check feature is enabled, BGP will accept incoming IP packets with a TTL value that is equal to or greater than the expected TTL value. Packets that are not accepted are discarded.  
  • The example configuration sets the expected incoming TTL value to at least 253, which is 255 minus the TTL value of 2, and this is the minimum TTL value expected from the BGP peer. The local router will accept the peering session from the 10.1.1.1 neighbor only if it is one or two hops away. |
| **Step 6** end | Exits configuration mode and enters privileged EXEC mode.  
  ```
  Example:
  Router(config-router)# end
  ``` |
### Command or Action

**Step 7** show running-config

**Example:**

```
Router# show running-config |
begin bgp
```

(Optional) Displays the contents of the currently running configuration file.

- The output of this command displays the configuration of the **neighbor ttl-security** command for each peer under the BGP configuration section of output. That section includes the neighbor address and the configured hop count.

**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

### Command or Action

**Step 8** show ip bgp neighbors [ip-address]

**Example:**

```
Router# show ip bgp neighbors 10.4.9.5
```

(Optional) Displays information about the TCP and BGP connections to neighbors.

- This command displays "External BGP neighbor may be up to number hops away" when the BGP Support for TTL Security Check feature is enabled. The **number** value represents the hop count. It is a number from 1 to 254.

**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

### Examples

The configuration of the BGP Support for TTL Security Check feature can be verified with the **show running-config** and **show ip bgp neighbors** commands. This feature is configured locally on each peer, so there is no remote configuration to verify.

The following is sample output from the **show running-config** command. The output shows that neighbor 10.1.1.1 is configured to establish or maintain the neighbor session only if the expected TTL count in the incoming IP packet is 253 or 254.

```
Router# show running-config |
begin bgp
router bgp 65000
   no synchronization
   bgp log-neighbor-changes
eighbor 10.1.1.1 remote-as 55000
neighbor 10.1.1.1 ttl-security hops 2
   no auto-summary
   .
   .
```

The following is sample output from the **show ip bgp neighbors** command. The output shows that the local router will accept packets from the 10.1.1.1 neighbor if it is no more than 2 hops away. The configuration of this feature is displayed in the address family section of the output. The relevant line is shown in bold in the output.

```
Router# show ip bgp neighbors 10.1.1.1
BGP neighbor is 10.1.1.1, remote AS 55000, external link
   BGP version 4, remote router ID 10.2.2.22
   BGP state - Established, up for 00:59:21
   Last read 00:00:21, hold time is 180, keepalive interval is 60 seconds
   Neighbor capabilities:
      Route refresh: advertised and received(new)
   Address family IPv4 Unicast: advertised and received
   Message statistics:
      InQ depth is 0
```
OutQ depth is 0

<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>226</td>
<td>227</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>228</td>
<td>229</td>
</tr>
</tbody>
</table>

Default minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast

BGP table version 1, neighbor version 1/0

Output queue sizes : 0 self, 0 replicated

Index 1, Offset 0, Mask 0x2

Member of update-group 1

Prefix activity: 

<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>n/a</td>
<td>0</td>
</tr>
</tbody>
</table>

Local Policy Denied Prefixes: 

<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of NLRIs in the update sent: max 0, min 0

Connections established 2; dropped 1

Last reset 00:59:50, due to User reset

External BGP neighbor may be up to 2 hops away.

Connection state is ESTAB, I/O status: 1, unread input bytes: 0

Local host: 10.2.2.22, Local port: 179

Foreign host: 10.1.1.1, Foreign port: 11001

Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)

Event Timers (current time is 0xCC28EC):

<table>
<thead>
<tr>
<th>Timer</th>
<th>Starts</th>
<th>Wakeups</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrans</td>
<td>63</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>TimeWait</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>AckHold</td>
<td>62</td>
<td>50</td>
<td>0x0</td>
</tr>
<tr>
<td>SendWnd</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>KeepAlive</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>GiveUp</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>PmtuAger</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>DeadWait</td>
<td>0</td>
<td>0</td>
<td>0x0</td>
</tr>
</tbody>
</table>

iss: 712702676 snduna: 712703881 sndnxt: 712703881 sndwnd: 15180
irs: 2255946817 rcvnxt: 2255948041 rcvwnd: 15161 delrcvwnd: 1223

minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms

Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):

Rcvd: 76 (out of order: 0), with data: 63, total data bytes: 1223

Sent: 113 (retransmit: 0, fastretransmit: 0), with data: 62, total data bytes: 4

Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following tasks:

- Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 276
- Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 279
- Enabling TCP Path MTU Discovery Globally for All BGP Sessions, page 281
- Enabling TCP Path MTU Discovery for a Single BGP Neighbor, page 283

Disabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to disable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but we recommend that you enter the show ip bgp neighbors command to ensure that TCP path MTU discovery is enabled.
This task assumes that you have previously configured BGP neighbors with active TCP connections.

**SUMMARY STEPS**

1. enable
2. show ip bgp neighbors [ip-address]
3. configure terminal
4. router bgp autonomous-system-number
5. no bgp transport path-mtu-discovery
6. end
7. show ip bgp neighbors [ip-address]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> show ip bgp neighbors [ip-address]</td>
<td>(Optional) Displays information about the TCP and BGP connections to neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp neighbors</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> no bgp transport path-mtu-discovery</td>
<td>Disables TCP path MTU discovery for all BGP sessions.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# no bgp transport path-mtu-discovery</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
Step 6  end | Exits router configuration mode and returns to privileged EXEC mode.
Example: | 
Router(config-router)# end
Step 7  show ip bgp neighbors [ip-address] | (Optional) Displays information about the TCP and BGP connections to neighbors.
Example: | 
Router# show ip bgp neighbors

#### Examples

The following sample output from the `show ip bgp neighbors` command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path mtu capable—show that TCP path MTU discovery is enabled.

Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2, remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
  .
  .
  .
  For address family: IPv4 Unicast
    BGP table version 5, neighbor version 5/0
    .
    .
    .
    Address tracking is enabled, the RIB does have a route to 172.16.1.2
    Address tracking requires at least a /24 route to the peer
    Last reset 00:00:35, due to Router ID changed
    Transport(tcp) path-mtu-discovery is enabled
    .
    .
    SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
    minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
    Flags: higher precedence, retransmission timeout, nagle, path mtu capable

The following is sample output from the `show ip bgp neighbors` command after the `no bgp transport path-mtu-discovery` command has been entered. Note that the path mtu entries are missing.

Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2, remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
  .
  .
  .
  For address family: IPv4 Unicast
    BGP table version 5, neighbor version 5/0
    .
    .
    .
    Address tracking is enabled, the RIB does have a route to 172.16.1.2
Address tracking requires at least a /24 route to the peer
Connections established 3; dropped 2
Last reset 00:00:35, due to Router ID changed

SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle

Disabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an internal BGP (iBGP) neighbor and then disable TCP path MTU discovery for the BGP neighbor session. The `neighbor transport` command can be used in router configuration or address family configuration mode.

This task assumes that you know that TCP path MTU discovery is enabled by default for all your BGP neighbors.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]}
5. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
6. neighbor {ip-address|peer-group-name} activate
7. no neighbor {ip-address|peer-group-name} transport {connection-mode | path-mtu-discovery}
8. end
9. show ip bgp neighbors

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family {ipv4</td>
<td>mdt</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td>• The example creates an IPv4 unicast address family session.</td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor [ip-address</td>
<td>peer-group-name] remote-as autonomous-system-number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor [ip-address</td>
<td>peer-group-name] activate</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 172.16.1.1 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> no neighbor [ip-address</td>
<td>peer-group-name] transport [connection-mode</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• In this example, TCP path MTU discovery is disabled for the neighbor at 172.16.1.1.</td>
</tr>
<tr>
<td>Router(config-router-af)# no neighbor 172.16.1.1 transport path-mtu-discovery</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
</tbody>
</table>
Command or Action | Purpose
---|---
Step 9 show ip bgp neighbors | (Optional) Displays information about the TCP and BGP connections to neighbors.

- In this example, the output from this command will not display that the neighbor has TCP path MTU discovery enabled.

Note: Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

### Examples

The following sample output shows that TCP path MTU discovery has been disabled for BGP neighbor 172.16.1.1 but that it is still enabled for BGP neighbor 192.168.2.2. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path mtu capable—show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.1, remote AS 45000, internal link
  BGP version 4, remote router ID 172.17.1.99
  Address tracking is enabled, the RIB does have a route to 172.16.1.1
  Address tracking requires at least a /24 route to the peer
  Connections established 1; dropped 0
  Last reset never
  SRTT: 165 ms, RTTO: 1172 ms, RTV: 1007 ms, KRTT: 0 ms
  minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
  Flags: higher precedence, retransmission timeout, nagle
  ...
  BGP neighbor is 192.168.2.2, remote AS 50000, external link
    BGP version 4, remote router ID 10.2.2.99
    For address family: IPv4 Unicast
      BGP table version 4, neighbor version 4/0
      Address tracking is enabled, the RIB does have a route to 192.168.2.2
      Address tracking requires at least a /24 route to the peer
      Connections established 2; dropped 1
      Last reset 00:05:11, due to User reset
      Transport(tcp) path-mtu-discovery is enabled
      ...
      SRTT: 210 ms, RTTO: 904 ms, RTV: 694 ms, KRTT: 0 ms
      minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
      Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

### Enabling TCP Path MTU Discovery Globally for All BGP Sessions

This section provides instructions on how to enable TCP path MTU discovery for all BGP sessions in the network:

1. **Step 9**: Use the `show ip bgp neighbors` command to display information about TCP and BGP connections to neighbors. This command is optional but useful for verifying that TCP path MTU discovery is enabled for all BGP neighbors.

2. **Example**: The output from the `show ip bgp neighbors` command for two BGP neighbors is shown. The example highlights that TCP path MTU discovery is enabled for both neighbors.

3. **Note**: The syntax applicable to this task is used in the example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Perform this task to enable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but if the BGP Support for TCP Path MTU Discovery per Session feature has been disabled, you can use this task to reenable it. To verify that TCP path MTU discovery is enabled, use the `show ip bgp neighbors` command.

This task assumes that you have previously configured BGP neighbors with active TCP connections.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `bgp transport path-mtu-discovery`
5. `end`
6. `show ip bgp neighbors`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp</code> <strong>autonomous-system-number</strong></td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>bgp transport path-mtu-discovery</code></td>
<td>Enables TCP path MTU discovery for all BGP sessions.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# bgp transport path-mtu-discovery</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 5  end</strong></td>
<td>Exits router configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# end</td>
</tr>
<tr>
<td><strong>Step 6  show ip bgp neighbors</strong></td>
<td>(Optional) Displays information about the TCP and BGP connections to neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show ip bgp neighbors</td>
</tr>
</tbody>
</table>

### Examples

The following sample output from the `show ip bgp neighbors` command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
  
  For address family: IPv4 Unicast
    BGP table version 5, neighbor version 5/0
    
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
    Address tracking requires at least a /24 route to the peer
    Connections established 3; dropped 2
    Last reset 00:00:35, due to Router ID changed
    Transport(tcp) path-mtu-discovery is enabled
  
  SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
  minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
  Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

### Enabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an eBGP neighbor and then enable TCP path MTU discovery for the BGP neighbor session. The `neighbor transport` command can be used in router configuration or address family configuration mode.
### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpng4 [unicast]}
5. neighbor {ip-address| peer-group-name} remote-as autonomous-system-number
6. neighbor {ip-address| peer-group-name} activate
7. neighbor {ip-address| peer-group-name} transport {connection-mode | path-mtu-discovery}
8. end
9. show ip bgp neighbors [ip-address]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family {ipv4 [mdt</td>
<td>multicast</td>
</tr>
<tr>
<td>Example:</td>
<td>• The example creates an IPv4 unicast address family session.</td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring BGP Neighbor Session Options

#### Enabling TCP Path MTU Discovery for a Single BGP Neighbor

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> neighbor `{ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.2.2 remote-as 50000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor `{ip-address</td>
<td>peer-group-name} activate`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.2.2 activate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor `{ip-address</td>
<td>peer-group-name} transport{connection-mode</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 192.168.2.2 transport path-mtu-discovery</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> <code>show ip bgp neighbors [ip-address]</code></td>
<td>(Optional) Displays information about the TCP and BGP connections to neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp neighbors 192.168.2.2</td>
<td></td>
</tr>
</tbody>
</table>

---

### Examples

The following sample output from the `show ip bgp neighbors` command shows that TCP path MTU discovery is enabled for the BGP neighbor at 192.168.2.2. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path-mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors 192.168.2.2
BGP neighbor is 192.168.2.2, remote AS 50000, external link
  BGP version 4, remote router ID 10.2.2.99
  ...
  ...
  For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
  ...
```

---

**Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*. 
Implementing BGP Dynamic Neighbors Using Subnet Ranges

In Cisco IOS XE Release 3.1S, support for BGP dynamic neighbors was introduced. Perform this task to implement the dynamic creation of BGP neighbors using subnet ranges.

In this task, a BGP peer group is created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer group is added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured. The peer group is activated under the IPv4 address family.

The next step is to move to another router--Router E in the figure below--where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.3.2) is within the configured subnet range for dynamic BGP peers. The task moves back to the first router, Router B, to run three `show` commands that have been modified to display dynamic BGP peer information.

This task requires Cisco IOS XE Release 3.1S, or a later release, to be running.
### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. bgp log-neighbor-changes
5. neighbor peer-group-name peer-group
6. bgp listen [limit max-number]
7. bgp listen [limit max-number | range network | length peer-group peer-group-name]
8. neighbor peer-group-name remote-as autonomous-system-number [alternate-as autonomous-system-number...]
9. address-family ipv4 [mdt | multicast | unicast [vrf vrf-name]]
10. neighbor {ip-address|peer-group-name} activate
11. end

12. Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.
13. enable
14. configure terminal
15. router bgp autonomous-system-number
16. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number [alternate-as autonomous-system-number...]

17. Return to the first router.
18. show ip bgp summary
19. show ip bgp peer-group [peer-group-name] [summary]
20. show ip bgp neighbors [ip-address]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterB&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>• The configuration is entered on router B.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterB# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring BGP Neighbor Session Options

**Enabling TCP Path MTU Discovery for a Single BGP Neighbor**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RouterB(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 4** bgp log-neighbor-changes | (Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. |
| **Example:** | |
| RouterB(config-router)# bgp log-neighbor-changes | |

| **Step 5** neighbor peer-group-name peer-group | Creates a BGP peer group. |
| **Example:** | |
| RouterB(config-router)# neighbor group192 peer-group | |

| **Step 6** bgp listen [limit max-number] | Sets a global limit of BGP dynamic subnet range neighbors. |
| **Example:** | |
| RouterB(config-router)# bgp listen limit 200 | |

**Note** Only the syntax applicable to this task is used in this example. For the complete syntax, see Step 7.

| **Step 7** bgp listen [limit max-number | range network | length peer-group peer-group-name] | Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature. |
| **Example:** | |
| RouterB(config-router)# bgp listen range 192.168.0.0/16 peer-group group192 | |

- Use the optional **limit** keyword and **max-number** argument to define the maximum number of BGP dynamic subnet range neighbors that can be created.
- Use the optional **range** keyword and **network | length** argument to define a prefix range to be associated with the specified peer group.
### Command or Action

#### Step 8

**neighbor peer-group-name remote-as autonomous-system-number [alternate-as autonomous-system-number...]**

Example:

```
RouterB(config-router)# neighbor group192 remote-as 40000 alternate-as 50000
```

Purpose

Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

- Use the optional `alternate-as` keyword and `autonomous-system-number` argument to identify up to five alternate autonomous system numbers for listen range neighbors.
- In this example, the peer group named group192 is configured with two possible autonomous system numbers.

**Note** The `alternate-as` keyword is used only with the listen range peer groups, not with individual BGP neighbors.

#### Step 9

**address-family ipv4 [mdt | multicast | unicast [vrf vrf-name]]**

Example:

```
RouterB(config-router)# address-family ipv4 unicast
```

Purpose

Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.

- The example creates an IPv4 unicast address family session.

#### Step 10

**neighbor {ip-address|peer-group-name} activate**

Example:

```
RouterB(config-router-af)# neighbor group192 activate
```

Purpose

Activates the neighbor or listen range peer group for the configured address family.

**Note** Usually BGP peer groups cannot be activated using this command, but the listen range peer groups are a special case.

#### Step 11

**end**

Example:

```
RouterB(config-router-af)# end
```

Purpose

Exits address family configuration mode and returns to privileged EXEC mode.

#### Step 12

Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.

#### Step 13

**enable**

Example:

```
RouterE> enable
```

Purpose

Enables privileged EXEC mode.

- Enter your password if prompted.
- The configuration is entered on Router E.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterE# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterE(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as autonomous-system-number [alternate-as autonomous-system-number... ]</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterE(config-router)# neighbor 192.168.3.1 remote-as 45000</td>
<td>• In this example, the interface (192.168.3.2 in the figure above) at Router E is with the subnet range set for the BGP listen range group, group192. When TCP opens a session to peer to Router B, Router B creates this peer dynamically.</td>
</tr>
<tr>
<td><strong>Step 17</strong> Return to the first router.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 18</strong> show ip bgp summary</td>
<td>(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterB# show ip bgp summary</td>
<td>• In this step, the configuration has returned to Router B.</td>
</tr>
<tr>
<td><strong>Step 19</strong> show ip bgp peer-group [peer-group-name] [summary]</td>
<td>(Optional) Displays information about BGP peer groups.</td>
</tr>
<tr>
<td><strong>Example:</strong> RouterB# show ip bgp peer-group group192</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 20**  
show ip bgp neighbors [ip-address]

### Purpose

(Optional) Displays information about BGP and TCP connections to neighbors.

- In this example, information is displayed about the dynamically created neighbor at 192.168.3.2. The IP address of this BGP neighbor can be found in the output of either the `show ip bgp summary` or the `show ip bgp peer-group` command.

**Note**  
Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference.

### Examples

The following output examples were taken from Router B in the figure above after the appropriate configuration steps in this task were completed on both Router B and Router E.

The following output from the `show ip bgp summary` command shows that the BGP neighbor 192.168.3.2 was dynamically created and is a member of the listen range group, group192. The output also shows that the IP prefix range of 192.168.0.0/16 is defined for the listen range named group192.

```
Router# show ip bgp summary  
BGP router identifier 192.168.3.1, local AS number 45000  
BGP table version is 1, main routing table version 1  
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd  
*192.168.3.2    4 50000       2       2        0    0    0 00:00:37        0  
* Dynamically created based on a listen range command  
Dynamically created neighbors: 1/(200 max), Subnet ranges: 1  
BGP peergroup group192 listen range group members:  
192.168.0.0/16  
```

The following output from the `show ip bgp peer-group` command shows information about the listen range group, group192 that was configured in this task:

```
Router# show ip bgp peer-group group192  
BGP peer-group is group192, remote AS 40000  
BGP peergroup group192 listen range group members:  
192.168.0.0/16  
BGP version 4  
Default minimum time between advertisement runs is 30 seconds  
For address family: IPv4 Unicast  
BGP neighbor is group192, peer-group external, members:  
*192.168.3.2  
Index 0, Offset 0, Mask 0x0  
Update messages formatted 0, replicated 0  
Number of NLRIs in the update sent: max 0, min 0  
```

The following sample output from the `show ip bgp neighbors` command shows that the neighbor 192.168.3.2 is a member of the peer group, group192, and belongs to the subnet range group 192.168.0.0/16, which shows that this peer was dynamically created:

```
Router# show ip bgp neighbors 192.168.3.2  
BGP neighbor is *192.168.3.2, remote AS 50000, external link  
Member of peer-group group192 for session parameters  
Belongs to the subnet range group: 192.168.0.0/16  
BGP version 4, remote router ID 192.168.3.2  
BGP state = Established, up for 00:06:35  
Last read 00:00:33, last write 00:00:29, hold time is 180, keepalive intervals  
Neighbor capabilities:  
  Route refresh: advertised and received(new)  
```
Configuration Examples for BGP Neighbor Session Options

- Example Configuring Fast Session Deactivation for a BGP Neighbor, page 292
- Example Configuring Selective Address Tracking for Fast Session Deactivation, page 293
- Example Configuring BFD for a BGP IPv6 Neighbor, page 293
- Example Restart Session After Maximum Number of Prefixes From Neighbor Reached, page 293
- Examples Configuring Dual-AS Peering for Network Migration, page 293
- Example Configuring the TTL-Security Check, page 295
- Examples Configuring BGP Support for TCP Path MTU Discovery per Session, page 295
- Example Implementing BGP Dynamic Neighbors Using Subnet Ranges, page 296

Example Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on Router A and Router B to monitor and use fast peering session deactivation for the neighbor session between the two routers. Although fast peering session deactivation is not required at both routers in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

**Router A**

```plaintext
router bgp 40000
neighbor 192.168.1.1 remote-as 45000
neighbor 192.168.1.1 fall-over
end
```

**Router B**

```plaintext
router bgp 45000
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.1.2 fall-over
end
```
Example Configuring Selective Address Tracking for Fast Session Deactivation

The following example shows how to configure the BGP peering session to be reset if a route with a prefix of /28 or a more specific route to a peer destination is no longer available:

```plaintext
router bgp 45000
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.1.2 fall-over route-map CHECK-NBR
exit
ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28
route-map CHECK-NBR permit 10
  match ip address prefix-list FILTER28
end
```

Example Configuring BFD for a BGP IPv6 Neighbor

The following example configures FastEthernet interface 0/1 with the IPv6 address 2001:DB8:4:1::1. Bidirectional Forwarding Detection (BFD) is configured for the BGP neighbor at 2001:DB8:5:1::2. BFD will track forwarding path failure of the BGP neighbor and provide faster reconvergence time for BGP after a forwarding path failure.

```plaintext
ipv6 unicast-routing
ipv6 cef
interface fastethernet 0/1
  ipv6 address 2001:DB8:4:1::1/64
  bfd interval 500 min_rx 500 multiplier 3
  no shutdown
exit
router bgp 65000
  no bgp default ipv4-unicast
  address-family ipv6 unicast
  neighbor 2001:DB8:5:1::2 remote-as 65001
  neighbor 2001:DB8:5:1::2 fall-over bfd
end
```

Example Restart Session After Maximum Number of Prefixes From Neighbor Reached

The following example sets the maximum number of prefixes allowed from the neighbor at 192.168.6.6 to 2000 and configures the router to reestablish a peering session after 30 minutes if one has been disabled:

```plaintext
router bgp 101
  network 172.16.0.0
  neighbor 192.168.6.6 maximum-prefix 2000 restart 30
```

Examples Configuring Dual-AS Peering for Network Migration

The following examples show how to configure and verify this feature:

- Example Dual-AS Configuration, page 294
- Example Dual-AS Confederation Configuration, page 294
- Example Replace-AS Configuration, page 295
Example Dual-AS Configuration

The following examples show how this feature is used to merge two autonomous systems without interrupting peering arrangements with the customer network. The `neighbor local-as` command is configured to allow Router 1 to maintain peering sessions through autonomous system 40000 and autonomous system 45000. Router 2 is a customer router that runs a BGP routing process in autonomous system 50000 and is configured to peer with autonomous-system 45000.

**Router 1 in Autonomous System 40000 (Provider Network)**

```plaintext
teardown: Serial3/0
  ip address 10.3.3.11 255.255.255.0
  router bgp 40000
    no synchronization
    no bgp router-id
    neighbor 10.3.3.33 remote-as 50000
    neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```

**Router 1 in Autonomous System 45000 (Provider Network)**

```plaintext
teardown: Serial3/0
  ip address 10.3.3.11 255.255.255.0
  router bgp 45000
    bgp router-id 10.0.0.11
    neighbor 10.3.3.33 remote-as 50000
```

**Router 2 in Autonomous System 50000 (Customer Network)**

```plaintext
teardown: Serial3/0
  ip address 10.3.3.33 255.255.255.0
  router bgp 50000
    bgp router-id 10.0.0.3
    neighbor 10.3.3.11 remote-as 45000
```

After the transition is complete, the configuration on router 50000 can be updated to peer with autonomous system 40000 during a normal maintenance window or during other scheduled downtime:

```plaintext
  neighbor 10.3.3.11 remote-as 100
```

Example Dual-AS Confederation Configuration

The following example can be used in place of the Router 1 configuration in the "Example: Dual-AS Configuration" example. The only difference between these configurations is that Router 1 is configured to be part of a confederation.

```plaintext
teardown: Serial3/0/0
  ip address 10.3.3.11 255.255.255.0
  router bgp 65534
    no synchronization
    bgp confederation identifier 100
    bgp router-id 10.0.0.11
    neighbor 10.3.3.33 remote-as 50000
    neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```
Example Replace-AS Configuration

The following example strips private autonomous system 64512 from outbound routing updates for the 10.3.3.33 neighbor and replaces it with autonomous system 50000:

```plaintext
router bgp 64512
    neighbor 10.3.3.33 local-as 50000 no-prepend replace-as
```

Example Configuring the TTL-Security Check

The example configurations in this section show how to configure the BGP Support for TTL Security Check feature.

The following example uses the `trace` command to determine the hop count to an eBGP peer. The hop count number is displayed in the output for each networking device that IP packets traverse to reach the specified neighbor. In the following example, the hop count for the 10.1.1.1 neighbor is 1.

```plaintext
Router# trace ip 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.1.1
  1 10.1.1.1  0 msec *  0 msec
```

The following example sets the hop count to 2 for the 10.1.1.1 neighbor. Because the hop-count argument is set to 2, BGP will accept only IP packets with a TTL count in the header that is equal to or greater than 253.

```plaintext
Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
```

Examples Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following configuration examples:

- Example Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 295
- Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 295
- Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions, page 296
- Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor, page 296

**Example Disabling TCP Path MTU Discovery Globally for All BGP Sessions**

The following example shows how to disable TCP path MTU discovery for all BGP neighbor sessions. Use the `show ip bgp neighbors` command to verify that TCP path MTU discovery has been disabled.

```plaintext
enable
configure terminal
router bgp 45000
    no bgp transport path-mtu-discovery
end
show ip bgp neighbors
```

**Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor**
Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions

The following example shows how to enable TCP path MTU discovery for all BGP neighbor sessions. Use the show ip bgp neighbors command to verify that TCP path MTU discovery has been enabled.

```sh
enable
configure terminal
router bgp 45000
  bgp transport path-mtu-discovery
end
show ip bgp neighbors
```

Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor

The following example shows how to enable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2. Use the show ip bgp neighbors command to verify that TCP path MTU discovery has been enabled.

```sh
enable
configure terminal
router bgp 45000
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 transport path-mtu-discovery
end
show ip bgp neighbors 192.168.2.2
```

Example Implementing BGP Dynamic Neighbors Using Subnet Ranges

In Cisco IOS XE Release 3.1S, support for BGP dynamic neighbors was introduced. The following example configurations show how to implement BGP dynamic neighbors using subnet ranges.

In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured for one of the peer groups, group192. The subnet range peer groups and a standard BGP peer are then activated under the IPv4 address family.

The configuration moves to another router--Router A in the figure below--where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.1.2) is within the configured subnet range for dynamic BGP peers.

A third router--Router E in the figure below--also starts a BGP peering session with Router B. Router E is in the autonomous system 50000, which is the configured alternate autonomous system. Router B responds to the resulting TCP session by creating another dynamic BGP peer.
This example concludes with the output of the `show ip bgp summary` command entered on Router B.

**Figure 27  BGP Dynamic Neighbor Topology**

**Router B**

```
enable
cfg-term
router bgp 45000
bgp log-neighbor-changes
bgp listen limit 200
bgp listen range 172.21.0.0/16 peer-group group172
neighbor group172 peer-group
neighbor group172 remote-as 45000
neighbor group192 peer-group
neighbor group192 remote-as 40000 alternate-as 50000
neighbor 172.16.1.2 remote-as 45000
address-family ipv4 unicast
neighbor group172 activate
neighbor group192 activate
neighbor 172.16.1.2 activate
end
```

**Router A**

```
enable
cfg-term
router bgp 40000
neighbor 192.168.1.1 remote-as 45000
exit
```

**Router E**

```
enable
cfg-term
router bgp 50000
neighbor 192.168.3.1 remote-as 45000
exit
```
After both Router A and Router E are configured, the `show ip bgp summary` command is run on Router B. The output displays the regular BGP neighbor, 172.16.1.2, and the two BGP neighbors that were created dynamically when Router A and Router E initiated TCP sessions for BGP peering to Router B. The output also shows information about the configured listen range subnet groups.

```
BGP router identifier 192.168.3.1, local AS number 45000
BGP table version is 1, main routing table version 1
Neighbor          V   AS MsgRcvd MsgSent   TblVer InQ OutQ Up/Down  State/PfxRcd
172.16.1.2        4   45000 15      15        1    0    0 00:12:20        0
*192.168.1.2      4   40000  3       3        1    0    0 00:00:37        0
*192.168.3.2      4   50000  6       6        1    0    0 00:04:36        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 2/(200 max), Subnet ranges: 2
BGP peergroup group172 listen range group members:
  172.21.0.0/16
BGP peergroup group192 listen range group members:
  192.168.0.0/16
```

### Additional References

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>Overview of Cisco BGP conceptual information with links to all the individual BGP modules</td>
<td>&quot;Cisco BGP Overview&quot; module</td>
</tr>
<tr>
<td>Conceptual and configuration details for basic BGP tasks</td>
<td>&quot;Configuring a Basic BGP Network&quot; module</td>
</tr>
<tr>
<td>Conceptual and configuration details for advanced BGP tasks</td>
<td>&quot;Configuring Advanced BGP Features&quot; module</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
</tr>
<tr>
<td>Bidirectional Forwarding Detection configuration tasks</td>
<td><em>Cisco IOS XE IP Routing: BFD Configuration Guide</em></td>
</tr>
</tbody>
</table>

**Where to Go Next**

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" module.
- If you want to configure some advanced BGP features including BGP next-hop address tracking and route dampening, see the "Configuring Advanced BGP Features" module.
## Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT SAFI</td>
<td>MDT SAFI</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-BGP4-MIB</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1191</td>
<td>Path MTU Discovery</td>
</tr>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 1772</td>
<td>Application of the Border Gateway Protocol in the Internet</td>
</tr>
<tr>
<td>RFC 1773</td>
<td>Experience with the BGP Protocol</td>
</tr>
<tr>
<td>RFC 1774</td>
<td>BGP-4 Protocol Analysis</td>
</tr>
<tr>
<td>RFC 1930</td>
<td>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
</tbody>
</table>

## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
# Feature Information for Configuring BGP Neighbor Session Options

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Dynamic Neighbors</td>
<td>Cisco IOS XE Release 3.1S</td>
<td>BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups. After a subnet range is configured for a BGP peer group and a TCP session is initiated for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. The new BGP neighbor will inherit any configuration for the peer group. The following commands were introduced or modified by this feature: <code>bgp listen</code>, <code>debug ip bgp range</code>, <code>neighbor remote-as</code>, <code>show ip bgp neighbors</code>, <code>show ip bgp peer-group</code>, <code>show ip bgp summary</code>.</td>
</tr>
<tr>
<td>BGP Hide Local-Autonomous System</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Restart Session After Max-Prefix Limit</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Restart Session After Max-Prefix Limit feature enhanced the capabilities of the <code>neighbor maximum-prefix</code> command with the introduction of the <code>restart</code> keyword. This enhancement allows the network operator to configure the time interval at which a peering session is reestablished by a router when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified by this release: <code>neighbor maximum-prefix</code>, <code>show ip bgp neighbors</code>.</td>
</tr>
<tr>
<td>BGP Selective Address Tracking</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Selective Address Tracking feature introduced the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were modified by this feature: <code>bgp nexthop</code>, <code>neighbor fall-over</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Support for Dual AS</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Support for Dual AS Configuration for Network AS Migrations feature extended the functionality of the BGP Local-AS feature by providing additional autonomous system path customization configuration options. The configuration of this feature is transparent to customer peering sessions, allowing the provider to merge two autonomous systems without interrupting customer peering arrangements. Customer peering sessions can later be updated during a maintenance window or during other scheduled downtime. This feature was introduced on the Cisco ASR 1000 Series Routers. The following command was modified by this feature: <code>neighbor local-as</code>.</td>
</tr>
<tr>
<td>Configuration for Network AS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Migrations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGP Support for Fast Peering Session Deactivation</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Support for Fast Peering Session Deactivation feature introduced an event-driven notification system that allows a Border Gateway Protocol (BGP) process to monitor BGP peering sessions on a per-neighbor basis. This feature improves the response time of BGP to adjacency changes by allowing BGP to detect an adjacency change and deactivate the terminated session in between standard BGP scanning intervals. Enabling this feature improves overall BGP convergence. This feature was introduced on the Cisco ASR 1000 Series Routers. The following command was modified by this feature: <code>neighbor fall-over</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Support for TCP Path MTU</td>
<td>Cisco IOS XE Release 2.1</td>
<td>BGP support for TCP path maximum transmission unit (MTU) discovery introduced the ability for BGP to automatically discover the best TCP path MTU for each BGP session. The TCP path MTU is enabled by default for all BGP neighbor sessions, but you can disable, and subsequently enable, the TCP path MTU globally for all BGP sessions or for an individual BGP neighbor session. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified by this feature: <code>bgp transport</code>, <code>neighbor transport</code>, <code>show ip bgp neighbors</code>.</td>
</tr>
<tr>
<td>Discovery per Session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGP Support for TTL Security Check</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Support for TTL Security Check feature introduced a lightweight security mechanism to protect external Border Gateway Protocol (eBGP) peering sessions from CPU utilization-based attacks using forged IP packets. Enabling this feature prevents attempts to hijack the eBGP peering session by a host on a network segment that is not part of either BGP network or by a host on a network segment that is not between the eBGP peers. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified by this feature: <code>neighbor ttl-security</code>, <code>show ip bgp neighbors</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP IPv6 Client for Single-Hop BFD</td>
<td>Cisco IOS XE Release 3.3S</td>
<td>Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that use an IPv6 address. The following command was modified by this feature: <code>neighbor fall-over</code>.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring Internal BGP Features

This document describes how to configure internal Border Gateway Protocol (BGP) features. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations.

- Finding Feature Information, page 305
- Information About Internal BGP Features, page 305
- How To Configure Internal BGP Features, page 311
- Internal BGP Feature Configuration Examples, page 320
- Additional References, page 322
- Feature Information for Configuring Internal BGP Features, page 324

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Internal BGP Features

- BGP Routing Domain Confederation, page 305
- BGP Route Reflector, page 306
- BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer, page 309
- BGP VPLS Autodiscovery Support on Route Reflector, page 310
- BGP Route Dampening, page 310

BGP Routing Domain Confederation

One way to reduce the internal BGP (iBGP) mesh is to divide an autonomous system into multiple subautonomous systems and group them into a single confederation. To the outside world, the confederation looks like a single autonomous system. Each autonomous system is fully meshed within itself, and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have external BGP (eBGP) sessions, they exchange routing information as if they were iBGP peers. Specifically, the next hop, Multi Exit Discriminator (MED)
attribute, and local preference information is preserved. This feature allows you to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number.

**BGP Route Reflector**

BGP requires that all iBGP speakers be fully meshed. However, this requirement does not scale well when there are many iBGP speakers. Instead of configuring a confederation, another way to reduce the iBGP mesh is to configure a route reflector.

The figure below illustrates a simple iBGP configuration with three iBGP speakers (Routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both routers B and C. Routers B and C do not readvertise the iBGP learned route to other iBGP speakers because the routers do not pass on routes learned from internal neighbors to other internal neighbors, thus preventing a routing information loop.

**Figure 28** Three Fully Meshed iBGP Speakers

With route reflectors, all iBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an iBGP peer is configured to be a route reflector responsible for passing iBGP learned routes to a set of iBGP neighbors. In the figure below, Router B is configured as a route

---

*Information About Internal BGP Features*

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2
reflector. When the route reflector receives routes advertised from Router A, it advertises them to Router C, and vice versa. This scheme eliminates the need for the iBGP session between Routers A and C.

*Figure 29  Simple BGP Model with a Route Reflector*

The internal peers of the route reflector are divided into two groups: client peers and all the other routers in the autonomous system (nonclient peers). A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster*. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with iBGP speakers outside their cluster.
The figure below illustrates a more complex route reflector scheme. Router A is the route reflector in a cluster with routers B, C, and D. Routers E, F, and G are fully meshed, nonclient routers.

**Figure 30 More Complex BGP Route Reflector Model**

When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups allowing an easy and gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All the other iBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group or nonclient group. In a simple configuration, the backbone could be divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all the route reflectors will be fully meshed). The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

Usually a cluster of clients will have a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might
have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

- Route Reflector Mechanisms to Avoid Routing Loops, page 309

**Route Reflector Mechanisms to Avoid Routing Loops**

As the iBGP learned routes are reflected, routing information may loop. The route reflector model has the following mechanisms to avoid routing loops:

- Originator ID is an optional, nontransitive BGP attribute. It is a 4-byte attribute created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.

- Cluster-list is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, and vice versa, it appends the local cluster ID to the cluster list. If the cluster list is empty, a new cluster list is created. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster list, the advertisement is ignored.

- The use of `set` clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most `set` clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only `set` clause of an outbound route map that is acted upon is the `set ip next-hop` clause.

**BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer**

The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.

The use of `set` clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most `set` clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only `set` clause of an outbound route map on a route reflector (RR) that is acted upon is the `set ip next-hop` clause. The `set ip next-hop` clause is applied to reflected routes.

Configuring an RR with an outbound route map allows a network administrator to modify the next hop attribute for a reflected route. By configuring a route map with the `set ip next-hop` clause, the administrator puts the RR into the forwarding path, and can configure iBGP multipath load sharing to achieve load balancing. That is, the RR can distribute outgoing packets among multiple egress points. See the "Configuring iBGP Multipath Load Sharing" module.

**Caution**

Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.
BGP VPLS Autodiscovery Support on Route Reflector

In Cisco IOS XE Release 2.5, BGP VPLS Autodiscovery Support on Route Reflector was introduced. On the ASR1000, BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector. The route reflector reflects the VPLS prefixes to other provider edge (PE) routers so that the PEs do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP VPLS address family on the route reflector.

For an example of a route reflector configuration that can reflect VPLS prefixes, see the Example BGP VPLS Autodiscovery Support on Route Reflector, page 322.

BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.

Note

No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

• Route Dampening Minimizes Route Flapping, page 310
• BGP Route Dampening Terms, page 310

Route Dampening Minimizes Route Flapping

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

BGP Route Dampening Terms

The following terms are used when describing route dampening:

• Flap--A route whose availability alternates repeatedly.
• History state--After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
• Penalty--Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
• Damp state--In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
• Suppress limit--A route is suppressed when its penalty exceeds this limit. The default value is 2000.
• Half-life--Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.
• Reuse limit--As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
• Maximum suppress limit--This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevent the iBGP peers from having a higher penalty for routes external to the autonomous system.

How To Configure Internal BGP Features

For information on configuring features that apply to multiple IP routing protocols (such as redistributing routing information), see the "Configuring IP Routing Protocol-Independent Features" module.

• Configuring a Routing Domain Confederation, page 311
• Configuring a Route Reflector, page 312
• Configuring a Route Reflector Using a Route Map to Set Next Hop for iBGP Peer, page 312
• Adjusting BGP Timers, page 316
• Configuring the Router to Consider a Missing MED as Worst Path, page 317
• Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths, page 317
• Configuring the Router to Use the MED to Choose a Path in a Confederation, page 317
• Enabling BGP Route Dampening, page 318
• Monitoring and Maintaining BGP Route Dampening, page 318

Configuring a Routing Domain Confederation

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number. To configure a BGP confederation identifier, use the following command in router configuration mode:
Command | Purpose
--- | ---
Router(config-router)# **bgp confederation** identifier as-number | Configures a BGP confederation.

In order to treat the neighbors from other autonomous systems within the confederation as special eBGP peers, use the following command in router configuration mode:

Command | Purpose
--- | ---
Router(config-router)# **bgp confederation peers** as-number [as-number] | Specifies the autonomous systems that belong to the confederation.

For an alternative way to reduce the iBGP mesh, see "Configuring a Route Reflector, page 312."

### Configuring a Route Reflector

To configure a route reflector and its clients, use the following command in router configuration mode:

Command | Purpose
--- | ---
Router(config-router)# **neighbor** (ip-address | peer-group-name) route-reflector-client | Configures the local router as a BGP route reflector and the specified neighbor as a client.

If the cluster has more than one route reflector, configure the cluster ID by using the following command in router configuration mode:

Command | Purpose
--- | ---
Router(config-router)# **bgp cluster-id** cluster-id | Configures the cluster ID.

Use the `show ip bgp` command to display the originator ID and the cluster-list attributes.

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, the route reflector need not reflect routes to clients.

To disable client-to-client route reflection, use the `no bgp client-to-client reflection` command in router configuration mode:

Command | Purpose
--- | ---
Router(config-router)# **no bgp client-to-client reflection** | Disables client-to-client route reflection.

### Configuring a Route Reflector Using a Route Map to Set Next Hop for iBGP Peer

Perform this task on an RR to set a next hop for an iBGP peer. One reason to perform this task is when you want to make the RR the next hop for routes, so that you can configure iBGP load sharing. Create a route
map that sets the next hop to be the RR’s address, which will be advertised to the RR clients. The route map is applied only to outbound routes from the router to which the route map is applied.

**Caution**
Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.

**Note**
Do not use the `neighbor next-hop-self` command to modify the next hop attribute for an RR. Using the `neighbor next-hop-self` command on the RR will modify next hop attributes only for non-reflected routes and not the intended routes that are being reflected from the RR clients. To modify the next hop attribute when reflecting a route, use an outbound route map.

This task configures the RR (Router 2) in the scenario illustrated in the figure below. In this case, Router 1 is the iBGP peer whose routes’ next hop is being set. Without a route map, outbound routes from Router 1 would go to next hop Router 3. Instead, setting the next hop to the RR’s address will cause routes from Router 1 to go to the RR, and thus allow the RR to perform load balancing among Routers 3, 4, and 5.

* Figure 31 * Route Reflector Using a Route Map to Set Next Hop for an iBGP Peer
### SUMMARY STEPS

1. enable
2. configure terminal
3. route-map map-tag
4. set ip next-hop ip-address
5. exit
6. router bgp as-number
7. address-family ipv4
8. maximum-paths ibgp number
9. neighbor ip-address remote-as as-number
10. neighbor ip-address activate
11. neighbor ip-address route-reflector-client
12. neighbor ip-address route-map map-name out
13. Repeat Steps 12 through 14 for the other RR clients.
14. end
15. show ip bgp neighbors

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 route-map map-tag</td>
<td>Enters route map configuration mode to configure a route map.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The route map is created to set the next hop for the route reflector client.</td>
</tr>
<tr>
<td>Router(config)# route-map rr-out</td>
<td></td>
</tr>
<tr>
<td>Step 4 set ip next-hop ip-address</td>
<td>Specifies that for routes that are advertised where this route map is applied, the next-hop attribute is set to this IPv4 address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• For this task, we want to set the next hop to be the address of the RR.</td>
</tr>
<tr>
<td>Router(config-route-map)# set ip next-hop 10.2.0.1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Step 5 exit</td>
<td>Exits route-map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-route-map)# exit</td>
</tr>
<tr>
<td>Step 6 router bgp as-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# router bgp 100</td>
</tr>
<tr>
<td>Step 7 address-family ipv4</td>
<td>Enters address family configuration mode to configure BGP peers to accept address family specific configurations.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# address-family ipv4</td>
</tr>
<tr>
<td>Step 8 maximum-paths ibgp number</td>
<td>Controls the maximum number of parallel iBGP routes that can be installed in the routing table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router)# maximum-paths ibgp 5</td>
</tr>
<tr>
<td>Step 9 neighbor ip-address remote-as as-number</td>
<td>Adds an entry to the BGP neighbor table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 10.1.0.1 remote-as 100</td>
</tr>
<tr>
<td>Step 10 neighbor ip-address activate</td>
<td>Enables the exchange of information with the peer.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 10.1.0.1 activate</td>
</tr>
<tr>
<td>Step 11 neighbor ip-address route-reflector-client</td>
<td>Configures the local router as a BGP route reflector, and configures the specified neighbor as a route-reflector client.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor 10.1.0.1 route-reflector-client</td>
</tr>
</tbody>
</table>
## Adjusting BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the Cisco IOS XE software declares a peer dead. By default, the keepalive timer is 60 seconds, and the hold-time timer is 180 seconds. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated hold time and the configured keepalive time.

To adjust BGP timers for all neighbors, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# timers bgp keepalive holdtime</td>
<td>Adjusts BGP timers for all neighbors.</td>
</tr>
</tbody>
</table>

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# neighbor (ip-address</td>
<td>Sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group.</td>
</tr>
<tr>
<td>peer-group-name) timers keepalive holdtime</td>
<td></td>
</tr>
</tbody>
</table>

---

### Command or Action | Purpose
---|---
**Step 12** | neighbor **ip-address** route-map **map-name** **out**<br>Example:<br>Router(config-router-af)# neighbor 10.1.0.1 route-map **rr-out** **out**<br>Applies the route map to outgoing routes from this neighbor.<br>- Reference the route map you created in Step 3.

**Step 13** | Repeat Steps 12 through 14 for the other RR clients.<br>You will not be applying a route map to the other RR clients.

**Step 14** | end<br>Example:<br>Router(config-router-af)# end<br>Exits address family configuration mode and enters privileged EXEC mode.

**Step 15** | show ip bgp neighbors<br>Example:<br>Router# show ip bgp neighbors<br>(Optional) Displays information about the BGP neighbors, including their status as RR clients, and information about the route map configured.

---

### Adjusting BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the Cisco IOS XE software declares a peer dead. By default, the keepalive timer is 60 seconds, and the hold-time timer is 180 seconds. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated hold time and the configured keepalive time.

To adjust BGP timers for all neighbors, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# timers bgp keepalive holdtime</td>
<td>Adjusts BGP timers for all neighbors.</td>
</tr>
</tbody>
</table>

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# neighbor (ip-address</td>
<td>Sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group.</td>
</tr>
<tr>
<td>peer-group-name) timers keepalive holdtime</td>
<td></td>
</tr>
</tbody>
</table>
The timers configured for a specific neighbor or peer group override the timers configured for all BGP neighbors using the `timers bgp` router configuration command.

To clear the timers for a BGP neighbor or peer group, use the `no` form of the `neighbor timers` command.

### Configuring the Router to Consider a Missing MED as Worst Path

To configure the router to consider a path with a missing MED attribute as the worst path, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config-router)# bgp bestpath med missing-as-worst</code></td>
<td>Configures the router to consider a missing MED as having a value of infinity, making the path without a MED value the least desirable path.</td>
</tr>
</tbody>
</table>

### Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths

To configure the router to consider the MED value in choosing a path, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config-router)# bgp bestpath med confed</code></td>
<td>Configures the router to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.</td>
</tr>
</tbody>
</table>

The comparison between MEDs is made only if there are no external autonomous systems in the path (an external autonomous system is an autonomous system that is not within the confederation). If there is an external autonomous system in the path, then the external MED is passed transparently through the confederation, and the comparison is not made.

The following example compares route A with these paths:

- path= 65000 65004, med=2
- path= 65001 65004, med=3
- path= 65002 65004, med=4
- path= 65003 1, med=1

In this case, path 1 would be chosen if the `bgp bestpath med confed router configuration` command is enabled. The fourth path has a lower MED, but it is not involved in the MED comparison because there is an external autonomous system in this path.

### Configuring the Router to Use the MED to Choose a Path in a Confederation

To configure the router to use the MED to choose the best path from among paths advertised by a single subautonomous system within a confederation, use the following command in router configuration mode:
Enabling BGP Route Dampening

To enable BGP route dampening, use the following command in address family or router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# bgp dampening</td>
<td>Enables BGP route dampening.</td>
</tr>
</tbody>
</table>

To change the default values of various dampening factors, use the following command in address family or router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# bgp dampening half-life reuse suppress max-suppress [route-map map-name]</td>
<td>Changes the default values of route dampening factors.</td>
</tr>
</tbody>
</table>

Monitoring and Maintaining BGP Route Dampening

You can monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life. To display flap statistics, use the following commands as needed:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show ip bgp flap-statistics</td>
<td>Displays BGP flap statistics for all paths.</td>
</tr>
<tr>
<td>Router# show ip bgp flap-statistics regexp regexp</td>
<td>Displays BGP flap statistics for all paths that match the regular expression.</td>
</tr>
<tr>
<td>Router# show ip bgp flap-statistics filter-list access-list</td>
<td>Displays BGP flap statistics for all paths that pass the filter.</td>
</tr>
</tbody>
</table>
### Command | Purpose
---|---
Router# `show ip bgp flap-statistics ip-address mask` | Displays BGP flap statistics for a single entry.

Router# `show ip bgp flap-statistics ip-address mask longer-prefix` | Displays BGP flap statistics for more specific entries.

To clear BGP flap statistics (thus making it less likely that the route will be dampened), use the following commands as needed:

### Command | Purpose
---|---
Router# `clear ip bgp flap-statistics` | Clears BGP flap statistics for all routes.

Router# `clear ip bgp flap-statistics regexp regexp` | Clears BGP flap statistics for all paths that match the regular expression.

Router# `clear ip bgp flap-statistics filter-list list` | Clears BGP flap statistics for all paths that pass the filter.

Router# `clear ip bgp flap-statistics ip-address mask` | Clears BGP flap statistics for a single entry.

Router# `clear ip bgp flap-statistics ip-address mask longer-prefix` | Clears BGP flap statistics for all paths from a neighbor.

---

**Note**

The flap statistics for a route are also cleared when a BGP peer is reset. Although the reset withdraws the route, there is no penalty applied in this instance, even if route flap dampening is enabled.

Once a route is dampened, you can display BGP route dampening information, including the time remaining before the dampened routes will be unsuppressed. To display the information, use the following command:

### Command | Purpose
---|---
Router# `show ip bgp dampened-paths` | Displays the dampened routes, including the time remaining before they will be unsuppressed.

You can clear BGP route dampening information and unsuppress any suppressed routes by using the following command:

### Command | Purpose
---|---
Router# `clear ip bgp dampening ip-address network-mask` | Clears route dampening information and unsuppresses the suppressed routes.
Internal BGP Feature Configuration Examples

- Example BGP Confederation Configurations with Route Maps, page 320
- Examples BGP Confederation, page 320
- Example Route Reflector Using a Route Map to Set Next Hop for iBGP Peer, page 321
- Example BGP VPLS Autodiscovery Support on Route Reflector, page 322

Example BGP Confederation Configurations with Route Maps

This section contains an example of the use of a BGP confederation configuration that includes BGP communities and route maps. For more examples of how to configure a BGP confederation, see the section Examples BGP Confederation, page 320 in this chapter.

This example shows how BGP community attributes are used with a BGP confederation configuration to filter routes.

In this example, the route map named `set-community` is applied to the outbound updates to neighbor 172.16.232.50 and the local-as community attribute is used to filter the routes. The routes that pass access list 1 have the special community attribute value local-as. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers outside autonomous system 200.

```
routing bgp 65000
  network 10.0.1.0 route-map set-community
  bgp confederation identifier 200
  bgp confederation peers 65001
  neighbor 172.16.232.50 remote-as 100
  neighbor 172.16.233.2 remote-as 65001
  route-map set-community permit 10
  match ip address 1
  set community local-as
```

Examples BGP Confederation

The following is a sample configuration that shows several peers in a confederation. The confederation consists of three internal autonomous systems with autonomous system numbers 6001, 6002, and 6003. To the BGP speakers outside the confederation, the confederation looks like a normal autonomous system with autonomous system number 500 (specified via the `bgp confederation identifier` router configuration command).

In a BGP speaker in autonomous system 6001, the `bgp confederation peers` router configuration command marks the peers from autonomous systems 6002 and 6003 as special eBGP peers. Hence peers 172.16.232.55 and 172.16.232.56 will get the local preference, next hop, and MED unmodified in the updates. The router at 10.16.69.1 is a normal eBGP speaker and the updates received by it from this peer will be just like a normal eBGP update from a peer in autonomous system 6001.

```
routing bgp 6001
  bgp confederation identifier 500
  bgp confederation peers 6002 6003
  neighbor 172.16.232.55 remote-as 6002
  neighbor 172.16.232.56 remote-as 6003
  neighbor 10.16.69.1 remote-as 777
```
In a BGP speaker in autonomous system 6002, the peers from autonomous systems 6001 and 6003 are configured as special eBGP peers. 10.70.70.1 is a normal iBGP peer and 10.99.99.2 is a normal eBGP peer from autonomous system 700.

```
router bgp 6002
  bgp confederation identifier 500
  bgp confederation peers 6001 6003
  neighbor 10.70.70.1 remote-as 6002
  neighbor 172.16.232.57 remote-as 6001
  neighbor 172.16.232.56 remote-as 6003
  neighbor 10.99.99.2 remote-as 700
```

In a BGP speaker in autonomous system 6003, the peers from autonomous systems 6001 and 6002 are configured as special eBGP peers. 10.200.200.200 is a normal eBGP peer from autonomous system 701.

```
router bgp 6003
  bgp confederation identifier 500
  bgp confederation peers 6001 6002
  neighbor 172.16.232.57 remote-as 6001
  neighbor 172.16.232.55 remote-as 6002
  neighbor 10.200.200.200 remote-as 701
```

The following is a part of the configuration from the BGP speaker 10.200.200.205 from autonomous system 701 in the same example. Neighbor 172.16.232.56 is configured as a normal eBGP speaker from autonomous system 500. The internal division of the autonomous system into multiple autonomous systems is not known to the peers external to the confederation.

```
router bgp 701
  neighbor 172.16.232.56 remote-as 500
  neighbor 10.200.200.205 remote-as 701
```

### Example Route Reflector Using a Route Map to Set Next Hop for iBGP Peer

The following example is based on the figure above. Router 2 is the route reflector for the clients: Routers 1, 3, 4, and 5. Router 1 is connected to Router 3, but you don’t want Router 1 to forward traffic destined to AS 200 to use Router 3 as the next hop (and therefore use the direct link with Router 3); you want to direct the traffic to the RR, which can load share among Routers 3, 4, and 5.

This example configures the RR, Router 2. A route map named rr-out is applied to Router 1; the route map sets the next hop to be the RR at 10.2.0.1. When Router 1 sees that the next hop is the RR address, Router 1 forwards the routes to the RR. When the RR receives packets, it will automatically load share among the iBGP paths. A maximum of five iBGP paths are allowed.

**Router 2**

```
route-map rr-out
  set ip next-hop 10.2.0.1
  !
  interface gigabitethernet 0/0
    ip address 10.2.0.1 255.255.0.0
  router bgp 100
    address-family ipv4 unicast
    maximum-paths ibgp 5
    neighbor 10.1.0.1 remote-as 100
    neighbor 10.1.0.1 activate
    neighbor 10.1.0.1 route-reflector-client
    neighbor 10.1.0.1 route-map rr-out out
    !
    neighbor 10.3.0.1 remote-as 100
    neighbor 10.3.0.1 activate
    neighbor 10.3.0.1 route-reflector-client
    !
    neighbor 10.4.0.1 remote-as 100
```

Example BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge Route Reflector) is configured as a route reflector capable of reflecting VPLS prefixes. The VPLS address family is configured by `address-family l2vpn vpls` below.

```
neighbor 10.4.0.1 activate
neighbor 10.4.0.1 route-reflector-client
!
neighbor 10.5.0.1 remote-as 100
neighbor 10.5.0.1 activate
neighbor 10.5.0.1 route-reflector-client
end
```

Additional References

The following sections provide references related to configuring internal BGP features.

```
hostname PE-RR
!
router bgp 1
  bgp router-id 1.1.1.3
  no bgp default route-target filter
  bgp log-neighbor-changes
  neighbor iBGP_PEERS peer-group
  neighbor iBGP_PEERS remote-as 1
  neighbor iBGP_PEERS update-source Loopback1
  neighbor 1.1.1.1 peer-group iBGP_PEERS
  neighbor 1.1.1.2 peer-group iBGP_PEERS
!
address-family l2vpn vpls
  neighbor iBGP_PEERS send-community extended
  neighbor iBGP_PEERS route-reflector-client
  neighbor 1.1.1.1 peer-group iBGP_PEERS
  neighbor 1.1.1.2 peer-group iBGP_PEERS
exit-address-family
!
```

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>BGP overview</td>
<td>&quot;Cisco BGP Overview&quot; module</td>
</tr>
<tr>
<td>Basic BGP configuration tasks</td>
<td>&quot;Configuring a Basic BGP Network&quot; module</td>
</tr>
<tr>
<td>iBGP multipath load sharing</td>
<td>&quot;iBGP Multipath Load Sharing&quot; module</td>
</tr>
<tr>
<td>Connecting to a service provider</td>
<td>&quot;Connecting to a Service Provider Using External BGP&quot; module</td>
</tr>
<tr>
<td>Configuring features that apply to multiple IP routing protocols</td>
<td><em>Cisco IOS IP Routing: Protocol-Independent Configuration Guide</em></td>
</tr>
</tbody>
</table>
### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

---

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

---

### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1772</td>
<td>Application of the Border Gateway Protocol in the Internet</td>
</tr>
<tr>
<td>RFC 1773</td>
<td>Experience with the BGP Protocol</td>
</tr>
<tr>
<td>RFC 1774</td>
<td>BGP-4 Protocol Analysis</td>
</tr>
<tr>
<td>RFC 1930</td>
<td>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</td>
</tr>
<tr>
<td>RFC 2519</td>
<td>A Framework for Inter-Domain Route Aggregation</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
<tr>
<td>RFC 3392</td>
<td>Capabilities Advertisement with BGP-4</td>
</tr>
<tr>
<td>RFC 4271</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 4893</td>
<td>BGP Support for Four-octet AS Number Space</td>
</tr>
<tr>
<td>RFC 5396</td>
<td>Textual Representation of Autonomous system (AS) Numbers</td>
</tr>
<tr>
<td>RFC 5398</td>
<td>Autonomous System (AS) Number Reservation for Documentation Use</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
<tr>
<td>documentation and tools for troubleshooting and resolving technical issues</td>
<td></td>
</tr>
<tr>
<td>with Cisco products and technologies.</td>
<td></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can</td>
<td></td>
</tr>
<tr>
<td>subscribe to various services, such as the Product Alert Tool (accessed</td>
<td></td>
</tr>
<tr>
<td>from Field Notices), the Cisco Technical Services Newsletter, and Really</td>
<td></td>
</tr>
<tr>
<td>Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user</td>
<td></td>
</tr>
<tr>
<td>ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for Configuring Internal BGP Features

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

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

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring internal BGP features</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This document describes how to configure internal BGP features. BGP is an</td>
</tr>
<tr>
<td></td>
<td></td>
<td>interdomain routing protocol that is designed to provide loop-free routing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>between organizations. This feature was introduced on the Cisco ASR 1000 Series</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Routers. The following commands were modified by this feature: bgp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>confederation identifier, bgp confederation peers.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Outbound Route Map on Route Reflector to Set IP Next Hop</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.</td>
</tr>
<tr>
<td>BGP VPLS Autodiscovery Support on Route Reflector</td>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring Advanced BGP Features

This module describes configuration tasks for various advanced Border Gateway Protocol (BGP) features. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. This module contains tasks to configure BGP next-hop address tracking, BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability, route dampening, Bidirectional Forwarding Detection (BFD) support for BGP, and BGP MIB support.

- Finding Feature Information, page 327
- Prerequisites for Configuring Advanced BGP Features, page 327
- Restrictions for Configuring Advanced BGP Features, page 327
- Information About Configuring Advanced BGP Features, page 328
- How to Configure Advanced BGP Features, page 336
- Configuration Examples for Configuring Advanced BGP Features, page 365
- Where to Go Next, page 369
- Additional References, page 369
- Feature Information for Configuring Advanced BGP Features, page 370

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Configuring Advanced BGP Features

Before configuring advanced BGP features you should be familiar with the "Cisco BGP Overview" module and the "Configuring a Basic BGP Network" module.

Restrictions for Configuring Advanced BGP Features

A router that runs Cisco IOS XE software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.
Information About Configuring Advanced BGP Features

- BGP Version 4, page 328
- BGP Support for Next-Hop Address Tracking, page 328
- BGP Nonstop Forwarding Awareness, page 329
- BGP Route Dampening, page 333
- BFD for BGP, page 334
- BGP MIB Support, page 334

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS). For more details about configuring a basic BGP network, see the "Configuring a Basic BGP Network" module.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. For more details about connecting to external BGP peers, see the "Connecting to a Service Provider Using External BGP" module.

Although BGP is referred to as an exterior gateway protocol (EGP) many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions. For more details about internal BGP peers, see the "Configuring Internal BGP Features" chapter of the Cisco IOS IP Routing Configuration Guide.

Note

BGP requires more configuration than other routing protocols and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

BGP Support for Next-Hop Address Tracking

To configure BGP next-hop address tracking, you should understand the following concepts:

- BGP Next-Hop Address Tracking, page 328
- Default BGP Scanner Behavior, page 329
- Selective BGP Next-Hop Route Filtering, page 329
- BGP Next_Hop Attribute, page 329

BGP Next-Hop Address Tracking

The BGP next-hop address tracking feature is enabled by default when a supporting Cisco software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as
peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best-path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

**Default BGP Scanner Behavior**

BGP monitors the next hop of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. By default, the BGP scanner is used to poll the RIB for this information every 60 seconds. During the 60 second time period between scan cycles, Interior Gateway Protocol (IGP) instability or other network failures can cause black holes and routing loops to temporarily form.

**Selective BGP Next-Hop Route Filtering**

BGP selective next-hop route filtering was implemented as part of the BGP Selective Address Tracking feature to support BGP next-hop address tracking. Selective next-hop route filtering uses a route map to selectively define routes to help resolve the BGP next hop.

The ability to use a route map with the `bgp next hop` command allows the configuration of the length of a prefix that applies to the BGP Next_Hop attribute. The route map is used during the BGP bestpath calculation and is applied to the route in the routing table that covers the next-hop attribute for BGP prefixes. If the next-hop route fails the route map evaluation, the next-hop route is marked as unreachable.

This command is per address family, so different route maps can be applied for next-hop routes in different address families.

**Note**

Only `match ip address` and `match source-protocol` commands are supported in the route map. No `set` commands or other `match` commands are supported.

**BGP Next_Hop Attribute**

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

**BGP Nonstop Forwarding Awareness**

To configure BGP Nonstop Forwarding (NSF) awareness, you should understand the following concepts:

- Cisco NSF Routing and Forwarding Operation, page 330
- Cisco Express Forwarding for NSF, page 330
- BGP Graceful Restart for NSF, page 331
- BGP NSF Awareness, page 331
- BGP Graceful Restart per Neighbor, page 332
- BGP Peer Session Templates, page 332
Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, EIGRP, OSPF, and IS-IS have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

In this document, a networking device is said to be NSF-aware if it is running NSF-compatible software. A device is said to be NSF-capable if it has been configured to support NSF; therefore, it would rebuild routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the line cards with the new FIB information.

Note
Currently, EIGRP supports only NSF awareness.

Cisco Express Forwarding for NSF

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by CEF. CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version ("epoch") number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

Note
For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.
BGP Graceful Restart for NSF

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable or NSF-aware router has "graceful restart capability." Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peer(s) (NSF-aware peers) need to exchange the graceful restart capability in their OPEN messages, at the time of session establishment. If both the peers do not exchange the graceful restart capability, the session will not be graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the NSF-capable device. This functionality will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

BGP NSF Awareness

BGP support for NSF requires that neighbor routers are NSF-aware or NSF-capable. NSF awareness in BGP is also enabled by the graceful restart mechanism. A router that is NSF-aware functions like a router that is NSF-capable with one exception: an NSF-aware router is incapable of performing an SSO operation. However, a router that is NSF-aware is capable of maintaining a peering relationship with a NSF-capable neighbor during a NSF SSO operation, as well as holding routes for this neighbor during the SSO operation.

The BGP Nonstop Forwarding Awareness feature provides an NSF-aware router with the capability to detect a neighbor that is undergoing an SSO operation, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes. The deployment of BGP NSF awareness can minimize the effects of route-processor (RP) failure conditions and improve the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

NSF awareness for BGP is not enabled by default. The `bgp graceful-restart` command is used to globally enable NSF awareness on a router that is running BGP. NSF-aware operations are also transparent to the network operator and BGP peers that do not support NSF capabilities.
NSF awareness is enabled automatically in supported software images for Interior Gateway Protocols, such as EIGRP, IS-IS, and OSPF. In BGP, global NSF awareness is not enabled automatically and must be started by issuing the `bgp graceful-restart` command in router configuration mode.

### BGP Graceful Restart per Neighbor

The ability to enable or disable BGP graceful restart for every individual BGP neighbor was introduced. Three new methods of configuring BGP graceful restart for BGP peers, in addition to the existing global BGP graceful restart configuration, are now available. Graceful restart can be enabled or disabled for a BGP peer or a BGP peer group using the `neighbor ha-mode graceful-restart` command, or a BGP peer can inherit a graceful restart configuration from a BGP peer-session template using the `ha-mode graceful-restart` command.

Although BGP graceful restart is disabled by default, the existing global command enables graceful restart for all BGP neighbors regardless of their capabilities. The ability to enable or disable BGP graceful restart for individual BGP neighbors provides a greater level of control for a network administrator.

When the BGP graceful restart capability is configured for an individual neighbor, each method of configuring graceful restart has the same priority, and the last configuration instance is applied to the neighbor. For example, if global graceful restart is enabled for all BGP neighbors but an individual neighbor is subsequently configured as a member of a peer group for which the graceful restart is disabled, graceful restart is disabled for that neighbor.

The configuration of the restart and stale-path timers is available only with the global `bgp graceful-restart` command, but the default values are set when the `neighbor ha-mode graceful-restart` or `ha-mode graceful-restart` commands are configured. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

### BGP Peer Session Templates

Peer session templates are used to group and apply the configuration of general BGP session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template.

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A BGP neighbor can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template. A BGP neighbor can directly inherit only one session template and can indirectly inherit up to seven additional peer session templates.

Peer session templates support inheritance. A directly applied peer session template can directly or indirectly inherit configurations from up to seven peer session templates. So, a total of eight peer session templates can be applied to a neighbor or neighbor group.

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.
For more details about BGP peer session templates, see the "Configuring a Basic BGP Network" module. To use a BGP peer session template to enable or disable BGP graceful restart, see the section "Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates".

**BGP Route Dampening**

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.

---

**Note**

No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

---

**Minimizing Flapping**

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

**Understanding Route Dampening Terms**

The following terms are used when describing route dampening:

- **Flap**--A route whose availability alternates repeatedly.
- **History state**--After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- **Penalty**--Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
- **Damp state**--In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- **Suppress limit**--A route is suppressed when its penalty exceeds this limit. The default value is 2000.
• Half-life--Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.
• Reuse limit--As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
• Maximum suppress limit--This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevents the iBGP peers from having a higher penalty for routes external to the autonomous system.

**BFD for BGP**

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a marked decrease in reconvergence time.

One caveat exists for BFD: BFD and BGP graceful restart capability cannot both be configured on a router running BGP. If an interface goes down, BFD detects the failure and indicates that the interface cannot be used for traffic forwarding and the BGP session goes down, but graceful restart still allows traffic forwarding on platforms that support NSF even though the BGP session is down, allowing traffic forwarding using the interface that is down. Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing.

See also the "Configuring BGP Neighbor Session Options" chapter, the section "Configuring BFD for BGP IPv6 Neighbors."

For more details about BFD, see the "Bidirectional Forwarding Detection" module of the *Cisco IOS XE IP Routing: BFD Configuration Guide*, Release 2.3.

**BGP MIB Support**

The Management Information Base (MIB) that supports BGP is the CISCO-BGP4-MIB. In Cisco IOS XE Release 2.1 and later releases, the BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following sections describe the objects and notifications (traps) that are supported:

**BGP FSM Transition Change Support**

The cbgpRouteTable supports BGP Finite State Machine (FSM) transition state changes.

The cbgpFsmStateChange object allows you to configure SNMP notifications (traps) for all FSM transition state changes. This notification contains the following MIB objects:

- `bgpPeerLastError`
- `bgpPeerState`
- `cbgpPeerLastErrorTxt`
- `cbgpPeerPrevState`
The cbgpBackwardTransition object supports all BGP FSM transition state changes. This object is sent each time the FSM moves to either a higher or lower numbered state. This notification contains the following MIB objects:

- bgpPeerLastError
- bgpPeerState
- cbgpPeerLastErrorTxt
- cbgpPeerPrevState

The `snmp-server enable bgp traps` command allows you to enable the traps individually or together with the existing FSM backward transition and established state traps as defined in RFC 1657.

### BGP Route Received Route Support

The cbgpRouteTable object supports the total number of routes received by a BGP neighbor. The following MIB object is used to query the CISCO-BGP4-MIB for routes that are learned from individual BGP peers:

- `cbgpPeerAddrFamilyPrefixTable`

Routes are indexed by the address-family identifier (AFI) or subaddress-family identifier (SAFI). The prefix information displayed in this table can also viewed in the output of the `show ip bgp` command.

### BGP Prefix Threshold Notification Support

The `cbgpPrefixMaxThresholdExceed` and `cbgpPrefixMaxThresholdClear` objects were introduced to allow you to poll for the total number of routes received by a BGP peer.

The `cbgpPrefixMaxThresholdExceed` object allows you to configure SNMP notifications to be sent when the prefix count for a BGP session has exceeded the configured value. This notification is configured on a per address family basis. The prefix threshold is configured with the `neighbor maximum-prefix` command. This notification contains the following MIB objects:

- `cbgpPeerPrefixAdminLimit`
- `cbgpPeerPrefixThreshold`

The `cbgpPrefixMaxThresholdClear` object allows you to configure SNMP notifications to be sent when the prefix count drops below the clear trap limit. This notification is configured on a per address family basis. This notification contains the following objects:

- `cbgpPeerPrefixAdminLimit`
- `cbgpPeerPrefixClearThreshold`

Notifications are sent when the prefix count drops below the clear trap limit for an address family under a BGP session after the `cbgpPrefixMaxThresholdExceed` notification is generated. The clear trap limit is calculated by subtracting 5 percent from the maximum prefix limit value configured with the `neighbor maximum-prefix` command. This notification will not be generated if the session goes down for any other reason after the `cbgpPrefixMaxThresholdExceed` is generated.

### VPNv4 Unicast Address Family Route Support

The `cbgpRouteTable` object allows you to configure SNMP GET operations for VPNv4 unicast address-family routes.

The following MIB object allows you to query for multiple BGP capabilities (for example, route refresh, multiprotocol BGP extensions, and graceful restart):

- `cbgpPeerCapsTable`
The following MIB object allows you to query for IPv4 and VPNv4 address family routes:

- `cbgpPeerAddrFamilyTable`

Each route is indexed by peer address, prefix, and prefix length. This object indexes BGP routes by the AFI and then by the SAFI. The AFI table is the primary index, and the SAFI table is the secondary index. Each BGP speaker maintains a local Routing Information Base (RIB) for each supported AFI and SAFI combination.

**cbgpPeerTable Support**

The `cbgpPeerTable` has been modified to support the enhancements described in this document. The following new table objects are supported in the CISCO-BGP-MIB.my:

- `cbgpPeerLastErrorTxt`
- `cbgpPeerPrevState`

The following table objects are not supported. The status of these objects is listed as deprecated, and these objects are not operational:

- `cbgpPeerPrefixAccepted`
- `cbgpPeerPrefixDenied`
- `cbgpPeerPrefixLimit`
- `cbgpPeerPrefixAdvertised`
- `cbgpPeerPrefixSuppressed`
- `cbgpPeerPrefixWithdrawn`

---

**How to Configure Advanced BGP Features**

- Configuring BGP Next-Hop Address Tracking, page 336
- Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart, page 343
- Configuring BGP Route Dampening, page 358
- Decreasing BGP Convergence Time Using BFD, page 361
- Enabling BGP MIB Support, page 364

**Configuring BGP Next-Hop Address Tracking**

The tasks in this section show how configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about configuring route dampening, see the Configuring BGP Route Dampening, page 358.

- Disabling BGP Next-Hop Address Tracking, page 336
- Adjusting the Delay Interval for BGP Next-Hop Address Tracking, page 338
- Configuring BGP Selective Next-Hop Route Filtering, page 339

**Disabling BGP Next-Hop Address Tracking**

---
Perform this task to disable BGP next-hop address tracking. BGP next-hop address tracking is enabled by default under the IPv4 and VPNv4 address families. Disabling next hop address tracking may be useful if you the network has unstable IGP peers and route dampening is not resolving the stability issues. To reenable BGP next-hop address tracking, use the `bgp` command with the `trigger` and `enable` keywords.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpn4 [unicast]]`
5. `no bgp nexthop trigger enable`
6. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| **Example:** | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | Router# configure terminal |
| **Step 3** `router bgp autonomous-system-number` | Enters router configuration mode to create or configure a BGP routing process. |
| **Example:** | Router(config)# router bgp 64512 |
| **Step 4** `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpn4 [unicast]]` | Enter address family configuration mode to configure BGP peers to accept address family-specific configurations.  
• The example creates an IPv4 unicast address family session. |
| **Example:** | Router(config-router)# address-family ipv4 unicast |
### Adjusting the Delay Interval for BGP Next-Hop Address Tracking

Perform this task to adjust the delay interval between routing table walks for BGP next-hop address tracking.

You can increase the performance of this feature by tuning the delay interval between full routing table walks to match the tuning parameters for the Interior Gateway protocol (IGP). The default delay interval is 5 seconds. This value is optimal for a fast-tuned IGP. In the case of an IGP that converges more slowly, you can change the delay interval to 20 seconds or more, depending on the IGP convergence time.

BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

#### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [[mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]]
5. bgp nexthop trigger delay delay-timer
6. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5 no bgp nexthop trigger enable</td>
<td>Disables BGP next-hop address tracking.</td>
</tr>
<tr>
<td></td>
<td>• Next-hop address tracking is enabled by default for IPv4 and VPNv4 address family sessions.</td>
</tr>
<tr>
<td></td>
<td>• The example disables next-hop address tracking.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 end</td>
<td>Exits address-family configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-router-af)# no bgp nexthop trigger enable
```

Example:

```
Router(config-router-af)# end
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**  
  Router# configure terminal | |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode to create or configure a BGP routing process. |
| **Example:**  
  Router(config)# router bgp 64512 | |
| **Step 4** address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | Enter address family configuration mode to configure BGP peers to accept address family-specific configurations.  
  • The example creates an IPv4 unicast address family session. |
| **Example:**  
  Router(config-router)# address-family ipv4 unicast | |
| **Step 5** bgp nexthop trigger delay delay-timer | Configures the delay interval between routing table walks for next-hop address tracking.  
  • The time period determines how long BGP will wait before starting a full routing table walk after notification is received.  
  • The value for the delay-timer argument is a number from 1 to 100 seconds. The default value is 5 seconds.  
  • The example configures a delay interval of 20 seconds. |
| **Example:**  
  Router(config-router-af)# bgp nexthop trigger delay 20 | |
| **Step 6** end | Exits address-family configuration mode, and enters privileged EXEC mode. |
| **Example:**  
  Router(config-router-af)# end | |

**Configuring BGP Selective Next-Hop Route Filtering**

Perform this task to configure selective next-hop route filtering using a route map to filter potential next-hop routes. This task uses prefix lists and route maps to match IP addresses or source protocols and can be used to avoid aggregate addresses and BGP prefixes being considered as next-hop routes.

For more examples of how to use the `bgp nexthop` command, see the Configuring BGP Selective Next-Hop Route Filtering Examples, page 366.
### Note
Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast] vrf vrf-name
5. bgp nexthop route-map map-name
6. exit
7. exit
8. ip prefix-list list-name [seq seq-value] [deny network / length | permit network / length][ge ge-value] [le le-value]
9. route-map map-name [permit] [deny][sequence-number]
10. match ip address prefix-list prefix-list-name [prefix-list-name...]
11. exit
12. route-map map-name [permit] [deny][sequence-number]
13. end
14. show ip bgp [network] [network-mask]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**

| address-family ipv4 [unicast | multicast] vrf vrf-name |

**Example:**

```bash
Router(config-router)# address-family ipv4 unicast
```

**Purpose:** Specifies the IPv4 address family and enters address family configuration mode.

- The `unicast` keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the `unicast` keyword is not specified with the `address-family ipv4` command.
- The `multicast` keyword specifies IPv4 multicast address prefixes.
- The `vrf` keyword and `vrf-name` argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

**Step 5**

| bgp next-hop route-map map-name |

**Example:**

```bash
Router(config-router-af)# bgp next-hop route-map CHECK-NEXTHOP
```

**Purpose:** Permits a route map to selectively define routes to help resolve the BGP next hop.

- In this example the route map named CHECK-NEXTHOP is created.

**Step 6**

| exit |

**Example:**

```bash
Router(config-router-af)# exit
```

**Purpose:** Exits address family configuration mode and enters router configuration mode.

**Step 7**

| exit |

**Example:**

```bash
Router(config-router)# exit
```

**Purpose:** Exits router configuration mode and enters global configuration mode.

**Step 8**

| ip prefix-list list-name [seq seq-value] [deny network / length | permit network / length] [ge ge-value] [le le-value] |

**Example:**

```bash
Router(config)# ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25
```

**Purpose:** Creates a prefix list for BGP next-hop route filtering.

- Selective next-hop route filtering supports prefix length matching or source protocol matching on a per address-family basis.
- The example creates a prefix list named FILTER25 that permits routes only if the mask length is more than 25; this will avoid aggregate routes being considered as the next-hop route.

**Step 9**

| route-map map-name [permit| deny] [sequence-number] |

**Example:**

```bash
Router(config)# route-map CHECK-NEXTHOP deny 10
```

**Purpose:** Configures a route map and enters route map configuration mode.

- In this example, a route map named CHECK-NEXTHOP is created. If there is an IP address match in the following `match` command, the IP address will be denied.
### Command or Action

| Step 10 | match ip address prefix-list prefix-list-name  
|         | [prefix-list-name...]  

#### Purpose
Matches the IP addresses in the specified prefix list.

- Use the `prefix-list-name` argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified.

#### Example:
```
Router(config-route-map)# match ip address prefix-list FILTER25
```

### Step 11 exit

#### Purpose
Exits route map configuration mode and enters global configuration mode.

#### Example:
```
Router(config-route-map)# exit
```

### Step 12 route-map map-name [permit|deny]  

[sequence-number]

#### Purpose
Configures a route map and enters route map configuration mode.

- In this example, all other IP addresses are permitted by route map CHECK-NEXTHOP.

#### Example:
```
Router(config)# route-map CHECK-NEXTHOP permit 20
```

### Step 13 end

#### Purpose
Exits route map configuration mode and returns to privileged EXEC mode.

#### Example:
```
Router(config-route-map)# end
```

### Step 14 show ip bgp [network] [network-mask]

#### Purpose
Displays the entries in the BGP routing table.

- Enter this command to view the next-hop addresses for each route.

#### Example:
```
Router# show ip bgp
```

### Examples

The following example from the `show ip bgp` command shows the next-hop addresses for each route:

```
BGP table version is 7, local router ID is 172.17.1.99  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - incomplete  
* 10.1.1.0/24  192.168.1.2  0 0 40000 i  
* 10.2.2.0/24  192.168.3.2  0 0 50000 i  
*> 172.16.1.0/24  0.0.0.0  0 32768 i  
*> 172.17.1.0/24  0.0.0.0  0 32768
```
Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart

The tasks in this section show how to configure BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability. The first task enables BGP NSF globally for all BGP neighbors and suggests a few troubleshooting options. The second task describes how to adjust the BGP graceful restart timers although the default settings are optimal for most network deployments. The next three tasks demonstrate how to enable or disable BGP graceful restart for individual BGP neighbors including peer session templates and peer groups. The final task verifies the local and peer router configuration of BGP NSF.

- Enabling BGP Global NSF Awareness Using BGP Graceful Restart, page 343
- Configuring BGP NSF Awareness Timers, page 345
- Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates, page 347
- Enabling BGP Graceful Restart for an Individual BGP Neighbor, page 352
- Disabling BGP Graceful Restart for a BGP Peer Group, page 354
- Verifying the Configuration of BGP Nonstop Forwarding Awareness, page 357

Enabling BGP Global NSF Awareness Using BGP Graceful Restart

Perform this task to enable BGP NSF awareness globally for all BGP neighbors. BGP NSF awareness is part of the graceful restart mechanism and BGP NSF awareness is enabled by issuing the `bgp graceful-restart` command in router configuration mode. BGP NSF awareness allows NSF-aware routers to support NSF-capable routers during an SSO operation. NSF-awareness is not enabled by default and should be configured on all neighbors that participate in BGP NSF.

Note

The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

Note

Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see "BFD for BGP".

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. bgp graceful-restart [restart-time seconds] [stalepath-time seconds]
5. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 enable</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2 configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3 router bgp autonomous-system-number</strong></td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4 bgp graceful-restart [restart-time seconds] [stalepath-time seconds]</strong></td>
<td>Enables the BGP graceful restart capability and BGP NSF awareness.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# bgp graceful-restart</td>
<td>If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor.</td>
</tr>
<tr>
<td></td>
<td>• Use this command on the restarting router and all of its peers (NSF-capable and NSF-aware).</td>
</tr>
<tr>
<td><strong>Step 5 end</strong></td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# end</td>
<td></td>
</tr>
</tbody>
</table>

- Troubleshooting Tips, page 344
- What to Do Next, page 345

### Troubleshooting Tips

To troubleshoot the NSF feature, use the following commands in privileged EXEC mode, as needed:

- **debug ip bgp** --Displays open messages that advertise the graceful restart capability.
- **debug ip bgp event** --Displays graceful restart timer events, such as the restart timer and the stalepath timer.
- **debug ip bgp updates** --Displays sent and received EOR messages. The EOR message is used by the NSF-aware router to start the stalepath timer, if configured.
• **show ip bgp** --Displays entries in the BGP routing table. The output from this command will display routes that are marked as stale by displaying the letter "S" next to each stale route.

• **show ip bgp neighbor** --Displays information about the TCP and BGP connections to neighbor devices. When enabled, the graceful restart capability is displayed in the output of this command.

**What to Do Next**

If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset by issuing the **clear ip bgp** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the "Configuring a Basic BGP Network" module.

**Configuring BGP NSF Awareness Timers**

Perform this task to adjust the BGP graceful restart timers. There are two BGP graceful restart timers that can be configured. The optional **restart-time** keyword and **seconds** argument determine how long peer routers will wait to delete stale routes before a BGP open message is received. The default value is 120 seconds. The optional **stalepath-time** keyword and **seconds** argument determine how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. The default value is 360 seconds.

The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. bgp graceful-restart [restart-time seconds]
5. bgp graceful-restart [stalepath-time seconds]
6. Router(config-router)# end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router> enable
```
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
</tbody>
</table>
| 4    | `bgp graceful-restart [restart-time]` | Enables the BGP graceful restart capability and BGP NSF awareness.  
  - The `restart-time` argument determines how long peer routers will wait to delete stale routes before a BGP open message is received.  
  - The default value is 120 seconds. The configurable range is from 1 to 3600 seconds.  
  
  **Note** Only the syntax applicable to this step is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference. |
| 5    | `bgp graceful-restart [stalepath-time]` | Enables the BGP graceful restart capability and BGP NSF awareness.  
  - The `stalepath-time` argument determines how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router.  
  - The default value is 360 seconds. The configurable range is from 1 to 3600 seconds.  
  
  **Note** Only the syntax applicable to this step is used in this example. For more details, see the Cisco IOS IP Routing: BGP Command Reference. |
| 6    | `Router(config-router)# end` | Exits router configuration mode and enters privileged EXEC mode. |

#### What to Do Next

If the `bgp graceful-restart` command has been issued after the BGP session has been established, you must reset the peer sessions by issuing the `clear ip bgp *` command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the `clear ip bgp` command, see the "Configuring a Basic BGP Network" module.
Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

Perform this task to enable and disable BGP graceful restart for BGP neighbors using peer session templates. In this task, a BGP peer session template is created, and BGP graceful restart is enabled. A second peer session template is created, and this template is configured to disable BGP graceful restart.

In this example, the configuration is performed at Router B in the figure below and two external BGP neighbors—at Router A and Router E in the figure below—are identified. The first BGP peer at Router A is configured to inherit the first peer session template that enables BGP graceful restart, whereas the second BGP peer at Router E inherits the second template that disables BGP graceful restart. Using the optional `show ip bgp neighbors` command, the status of the BGP graceful restart capability is verified for each BGP neighbor configured in this task.

The restart and stale-path timers can be modified only using the global `bgp graceful-restart` command as shown in the GUID-4B37FD3F-E95B-4BA6-909C-EF929A1F6D71. The restart and stale-path timers are set to the default values when BGP graceful restart is enabled for BGP neighbors using peer session templates.

**Note**

A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `template peer-session session-template-name`
5. `ha-mode graceful-restart [disable]`
6. `exit-peer-session`
7. `template peer-session session-template-name`
8. `ha-mode graceful-restart [disable]`
9. `exit-peer-session`
10. `bgp log-neighbor-changes`
11. `neighbor ip-address remote-as autonomous-system-number`
12. `neighbor ip-address inherit peer-session session-template-number`
13. `neighbor ip-address remote-as autonomous-system-number`
14. `neighbor ip-address inherit peer-session session-template-number`
15. `end`
16. `show ip bgp template peer-session [session-template-number]`
17. `show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]]]`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>
### Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4** template peer-session session-template-name | Enters session-template configuration mode and creates a peer session template.  
• In this example, a peer session template named S1 is created. |
| **Example:**  
Router(config-router)# template peer-session S1 | |
| **Step 5** ha-mode graceful-restart [disable] | Enables the BGP graceful restart capability and BGP NSF awareness.  
• Use the `disable` keyword to disable BGP graceful restart capability.  
• If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor.  
• In this example, the BGP graceful restart capability is enabled for the peer session template named S1. |
| **Example:**  
Router(config-router-stmp)# ha-mode graceful-restart | |
| **Step 6** exit-peer-session | Exits session-template configuration mode and returns to router configuration mode. |
| **Example:**  
Router(config-router-stmp)# exit-peer-session | |
| **Step 7** template peer-session session-template-name | Enters session-template configuration mode and creates a peer session template.  
• In this example, a peer session template named S2 is created. |
| **Example:**  
Router(config-router)# template peer-session S2 | |
| **Step 8** ha-mode graceful-restart [disable] | Enables the BGP graceful restart capability and BGP NSF awareness.  
• Use the `disable` keyword to disable BGP graceful restart capability.  
• If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor.  
• In this example, the BGP graceful restart capability is disabled for the peer session template named S2. |
| **Example:**  
Router(config-router-stmp)# ha-mode graceful-restart disable | |
| **Step 9** exit-peer-session | Exits session-template configuration mode and returns to router configuration mode. |
| **Example:**  
Router(config-router-stmp)# exit-peer-session | |
### Configuring Advanced BGP Features

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong> <code>bgp log-neighbor-changes</code></td>
<td>Enables logging of BGP neighbor status changes (up or down) and neighbor resets.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# bgp log-neighbor-changes</td>
</tr>
<tr>
<td></td>
<td>- Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.</td>
</tr>
<tr>
<td><strong>Step 11</strong> <code>neighbor ip-address remote-as autonomous-system-number</code></td>
<td>Configures peering with a BGP neighbor in the specified autonomous system.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</td>
</tr>
<tr>
<td></td>
<td>- In this example, the BGP peer at 192.168.1.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).</td>
</tr>
<tr>
<td><strong>Step 12</strong> <code>neighbor ip-address inherit peer-session session-template-number</code></td>
<td>Inherits a peer session template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# neighbor 192.168.1.2 inherit peer-session S1</td>
</tr>
<tr>
<td></td>
<td>- In this example, the peer session template named S1 is inherited, and the neighbor inherits the enabling of BGP graceful restart.</td>
</tr>
<tr>
<td><strong>Step 13</strong> <code>neighbor ip-address remote-as autonomous-system-number</code></td>
<td>Configures peering with a BGP neighbor in the specified autonomous system.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# neighbor 192.168.3.2 remote-as 50000</td>
</tr>
<tr>
<td></td>
<td>- In this example, the BGP peer at 192.168.3.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).</td>
</tr>
<tr>
<td><strong>Step 14</strong> <code>neighbor ip-address inherit peer-session session-template-number</code></td>
<td>Inherits a peer session-template.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# neighbor 192.168.3.2 inherit peer-session S2</td>
</tr>
<tr>
<td></td>
<td>- In this example, the peer session template named S2 is inherited, and the neighbor inherits the disabling of BGP graceful restart.</td>
</tr>
<tr>
<td><strong>Step 15</strong> <code>end</code></td>
<td>Exits router configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router)# end</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 16   | `show ip bgp template peer-session [session-template-number]` | (Optional) Displays locally configured peer session templates.  
- The output can be filtered to display a single peer policy template with the `session-template-name` argument. This command also supports all standard output modifiers. |

**Example:**
```
Router# show ip bgp template peer-session
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 17   | `show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy{detail}]]` | (Optional) Displays information about TCP and BGP connections to neighbors.  
- "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router.  
- In this example, the output is filtered to display information about the BGP peer at 192.168.1.2. |

**Example:**
```
Router# show ip bgp neighbors 192.168.1.2
```

### Examples

The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 192.168.1.2 (Router A in the figure above). Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can only be set using the `bgp graceful-restart` command.

```
Router# show ip bgp neighbors 192.168.1.2
BGP neighbor is 192.168.1.2, remote AS 40000, external link
Inherits from template S1 for session parameters
BGP version 4, remote router ID 192.168.1.2
BGP state = Established, up for 00:02:11
Last read 00:00:23, last write 00:00:27, hold time is 180, keepalive intervals
Neighbor sessions:
1 active, is multisession capable
Neighbor capabilities:
- Route refresh: advertised and received(new)
- Address family IPv4 Unicast: advertised and received
- Graceful Restart Capability: advertised
- Multisession Capability: advertised and received

Address tracking is enabled, the RIB does have a route to 192.168.1.2
Connections established 1; dropped 0
Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 192.168.3.2 (Router E in the figure above). Graceful restart is shown as disabled.

```
Router# show ip bgp neighbors 192.168.3.2
BGP neighbor is 192.168.3.2, remote AS 50000, external link
Inherits from template S2 for session parameters
BGP version 4, remote router ID 192.168.3.2
BGP state = Established, up for 00:01:41
Last read 00:00:45, last write 00:00:45, hold time is 180, keepalive intervals
Neighbor sessions:
1 active, is multisession capable
Neighbor capabilities:
```
Route refresh: advertised and received (new)
Address family IPv4 Unicast: advertised and received

Address tracking is enabled, the RIB does have a route to 192.168.3.2
Connections established: 1; dropped: 0
Last reset never
Transport (tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0

Enabling BGP Graceful Restart for an Individual BGP Neighbor

Perform this task on Router B in the figure above to enable BGP graceful restart on the internal BGP peer at Router C in the figure above. Under address family IPv4, the neighbor at Router C is identified, and BGP graceful restart is enabled for the neighbor at Router C with the IP address 172.21.1.2. To verify that BGP graceful restart is enabled, the optional `show ip bgp neighbors` command is used.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [unicast | multicast | vrf vrf-name]`
5. `neighbor ip-address remote-as autonomous-system-number`
6. `neighbor ip-address activate`
7. `neighbor ip-address ha-mode graceful-restart [disable]`
8. `end`
9. `show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics| received prefix-filter| policy[detail]]]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [unicast</td>
<td>multICAST</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 unicast</td>
<td>• The <strong>unicast</strong> keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the <strong>unicast</strong> keyword is not specified with the <strong>address-family ipv4</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>multicast</strong> keyword specifies IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>vrf</strong> keyword and <strong>vrf-name</strong> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Configures peering with a BGP neighbor in the specified autonomous system.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 172.21.1.2 remote-as 45000</td>
<td>• In this example, the BGP peer at 172.21.1.2 is an internal BGP peer because it has the same autonomous system number as the router where the BGP configuration is being entered (see Step 3).</td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address activate</td>
<td>Enables the neighbor to exchange prefixes for the IPv4 address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 172.21.1.2 activate</td>
<td>• In this example, the internal BGP peer at 172.21.1.2 is activated.</td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor ip-address ha-mode graceful-restart [disable]</td>
<td>Enables the BGP graceful restart capability for a BGP neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 172.21.1.2 ha-mode graceful-restart</td>
<td>• Use the <strong>disable</strong> keyword to disable BGP graceful restart capability.</td>
</tr>
<tr>
<td></td>
<td>• If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor.</td>
</tr>
<tr>
<td></td>
<td>• In this example, the BGP graceful restart capability is enabled for the neighbor at 172.21.1.2.</td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Router(config-router-af)# end | }
Command or Action | Purpose
---|---
**Step 9** show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [reexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]]]

(Optional) Displays information about TCP and BGP connections to neighbors.

- "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router.
- In this example, the output is filtered to display information about the BGP peer at 172.21.1.2.

**Examples**
The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 172.21.1.2. Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global `bgp graceful-restart` command.

```
Router# show ip bgp neighbors 172.21.1.2
BGP neighbor is 172.21.1.2, remote AS 45000, internal link
BGP version 4, remote router ID 172.22.1.1
BGP state = Established, up for 00:01:01
Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
Neighbor sessions:
  1 active, is multisession capable
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Address family IPv4 Unicast: advertised and received
  Graceful Restart Capability: advertised
  Multisession Capability: advertised and received
Address tracking is enabled, the RIB does have a route to 172.21.1.2
Connections established 1; dropped 0
Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Reset is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

**Disabling BGP Graceful Restart for a BGP Peer Group**
Perform this task to disable BGP graceful restart for a BGP peer group. In this task, a BGP peer group is created and graceful restart is disabled for the peer group. A BGP neighbor, 172.16.1.2 at Router D in the figure above, is then identified and added as a peer group member and inherits the configuration associated with the peer group, which, in this example, disables BGP graceful restart.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast] vrf vrf-name
5. neighbor peer-group-name peer-group
6. neighbor peer-group-name remote-as autonomous-system-number
7. neighbor peer-group-name ha-mode graceful-restart [disable]
8. neighbor ip-address peer-group peer-group-name
9. end
10. show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics| received prefix-filter| policy|detail]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
| | • Enter your password if prompted. |
| **Example:** | |
| Router> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | |
| Router# configure terminal | |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode and creates a BGP routing process. |
| **Example:** | |
| Router(config)# router bgp 45000 | |
| **Step 4** address-family ipv4 [unicast | multicast] vrf vrf-name] | Specifies the IPv4 address family and enters address family configuration mode. |
| | • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. |
| | • The **multicast** keyword specifies IPv4 multicast address prefixes. |
| | • The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. |
| **Example:** | |
| Router(config-router)# address-family ipv4 unicast | |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 5    | `neighbor peer-group-name peer-group` | Creates a BGP peer group.  
- In this example, the peer group named PG1 is created. |
|      | **Example:** |
|      | Router(config-router-af)# neighbor PG1 peer-group |
| 6    | `neighbor peer-group-name remote-as autonomous-system-number` | Configures peering with a BGP peer group in the specified autonomous system.  
- In this example, the BGP peer group named PG1 is added to the IPv4 multiprotocol BGP neighbor table of the local router. |
|      | **Example:** |
|      | Router(config-router-af)# neighbor PG1 remote-as 45000 |
| 7    | `neighbor peer-group-name ha-mode graceful-restart [disable]` | Enables the BGP graceful restart capability for a BGP neighbor.  
- Use the `disable` keyword to disable BGP graceful restart capability.  
- If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor.  
- In this example, the BGP graceful restart capability is disabled for the BGP peer group named PG1. |
|      | **Example:** |
|      | Router(config-router-af)# neighbor PG1 ha-mode graceful-restart disable |
| 8    | `neighbor ip-address peer-group peer-group-name` | Assigns the IP address of a BGP neighbor to a peer group.  
- In this example, the BGP neighbor peer at 172.16.1.2 is configured as a member of the peer group named PG1. |
|      | **Example:** |
|      | Router(config-router-af)# neighbor 172.16.1.2 peer-group PG1 |
| 9    | `end` | Exits address family configuration mode and returns to privileged EXEC mode. |
|      | **Example:** |
|      | Router(config-router-af)# end |
| 10   | `show ip bgp neighbors [ip-address] [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]]]` | (Optional) Displays information about TCP and BGP connections to neighbors.  
- In this example, the output is filtered to display information about the BGP peer at 172.16.1.2 and the "Graceful-Restart is disabled" line shows that the graceful restart capability is disabled for this neighbor. |
|      | **Example:** |
|      | Router# show ip bgp neighbors 172.16.1.2 |
Examples

The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 172.16.1.2. Graceful restart is shown as disabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global `bgp graceful-restart` command.

```
Router# show ip bgp neighbors 172.16.1.2
BGP neighbor is 172.16.1.2, remote AS 45000, internal link
Member of peer-group PG1 for session parameters
BGP version 4, remote router ID 0.0.0.0
BGP state = Idle
Neighbor sessions:
  0 active, is multisession capable
  !
Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Connections established 0; dropped 0
  Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
```

Verifying the Configuration of BGP Nonstop Forwarding Awareness

Use the following steps to verify the local configuration of BGP NSF awareness on a router and to verify the configuration of NSF awareness on peer routers in a BGP network.

**SUMMARY STEPS**

1. **enable**
2. **show running-config [options]**
3. **show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]])**

**DETAILED STEPS**

**Step 1**

**enable**

Enables privileged EXEC mode. Enter your password if prompted.

**Example:**

```
Router> enable
```

**Step 2**

**show running-config [options]**

Displays the running configuration on the local router. The output will display the configuration of the `bgp graceful-restart` command in the BGP section. Repeat this command on all BGP neighbor routers to verify that all BGP peers are configured for BGP NSF awareness. In this example, BGP graceful restart is enabled globally and the external neighbor at 192.168.1.2 is configured to be a BGP peer and will have the BGP graceful restart capability enabled.

**Example:**

```
Router# show running-config
```

```
router bgp 45000
bgp router-id 172.17.1.99
bgp log-neighbor-changes
```
bpg graceful-restart restart-time 130
bpg graceful-restart stalepath-time 350
bpg graceful-restart timers bgp 70 120
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.1.2 activate

Step 3 show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]]]
Displays information about TCP and BGP connections to neighbors. "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. In Cisco IOS XE Release 2.1 or later releases, the ability to enable or disable the BGP graceful restart capability for an individual BGP neighbor, peer group or peer session template was introduced and output was added to this command to show the BGP graceful restart status.

Configuring BGP Route Dampening
The tasks in this section configure and monitor BGP route dampening. Route dampening is designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

- Enabling and Configuring BGP Route Dampening, page 358
- Monitoring and Maintaining BGP Route Dampening, page 359

Enabling and Configuring BGP Route Dampening
Perform this task to enable and configure BGP route dampening.

SUMMARY STEPS
1. enable
2. configure terminal
3. router bgp as-number
4. address-family ipv4 [unicast | multicast | vrf vrf-name]
5. bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name]
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

Example:
Router> enable
## Command or Action | Purpose
--- | ---
Step 2 configure terminal | Enters global configuration mode.

**Example:**
```
Router# configure terminal
```

Step 3 router bgp as-number | Enters router configuration mode and creates a BGP routing process.

**Example:**
```
Router(config)# router bgp 45000
```

Step 4 address-family ipv4 [unicast | multicast] vrf vrf-name | Specifies the IPv4 address family and enters address family configuration mode.

- The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the `address-family ipv4` command.
- The **multicast** keyword specifies IPv4 multicast address prefixes.
- The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

**Example:**
```
Router(config-router)# address-family ipv4 unicast
```

Step 5 bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name] | Enables BGP route dampening and changes the default values of route dampening factors.

- The **half-life**, **reuse**, **suppress**, and **max-suppress-time** arguments are all position dependent; if one argument is entered then all the arguments must be entered.
- Use the **route-map** keyword and **map-name** argument to control where BGP route dampening is enabled.

**Example:**
```
Router(config-router-af)# bgp dampening 30 1500 10000 120
```

Step 6 end | Exits address family configuration mode and enters privileged EXEC mode.

**Example:**
```
Router(config-router-af)# end
```

## Monitoring and Maintaining BGP Route Dampening

Perform the steps in this task as required to monitor and maintain BGP route dampening.

---

**Monitoring and Maintaining BGP Route Dampening**

Perform the steps in this task as required to monitor and maintain BGP route dampening.
SUMMARY STEPS

1. enable
2. show ip bgp flap-statistics [regexp regexp | filter-list access-list | ip-address mask [longer-prefix]]
3. clear ip bgp flap-statistics [neighbor-address [ipv4-mask]] [regexp regexp | filter-list extcom-number]
4. show ip bgp dampened-paths
5. clear ip bgp [ipv4 {multicast | unicast} | ipv6{multicast | unicast} | vpnv4 unicast] dampening [neighbor-address] [ipv4-mask]

DETAILED STEPS

Step 1 enable
Enables privileged EXEC mode. Enter your password if prompted.

Example:
Router> enable

Step 2 show ip bgp flap-statistics [regexp regexp | filter-list access-list | ip-address mask [longer-prefix]]
Use this command to monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life.

Example:
Router# show ip bgp flap-statistics
BGP table version is 10, local router ID is 172.17.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network From Flaps Duration Reuse Path
*d 10.0.0.0 172.17.232.177 4 00:13:31 00:18:10 100
*d 10.2.0.0 172.17.232.177 4 00:02:45 00:28:20 100

Step 3 clear ip bgp flap-statistics [neighbor-address [ipv4-mask]] [regexp regexp | filter-list extcom-number]
Use this command to clear the accumulated penalty for routes that are received on a router that has BGP dampening enabled. If no arguments or keywords are specified, flap statistics are cleared for all routes. Flap statistics are also cleared when the peer is stable for the half-life time period. After the BGP flap statistics are cleared, the route is less likely to be dampened.

Example:
Router# clear ip bgp flap-statistics 172.17.232.177

Step 4 show ip bgp dampened-paths
Use this command to monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life.

Example:
Router# show ip bgp dampened-paths
BGP table version is 10, local router ID is 172.29.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Step 5  clear ip bgp [ipv4 {multicast | unicast} | ipv6 {multicast | unicast} | vpnv4 unicast] dampening [neighbor-address] [ipv4-mask]

Use this command to clear stored route dampening information. If no keywords or arguments are entered, route dampening information for the entire routing table is cleared. The following example clears route dampening information for VPNv4 address family prefixes from network 192.168.10.0/24, and unsuppresses its suppressed routes.

Example:

Router# clear ip bgp vpnv4 unicast dampening 192.168.10.0 255.255.255.0

Decreasing BGP Convergence Time Using BFD

You start a BFD process by configuring BFD on the interface. When the BFD process is started, no entries are created in the adjacency database, in other words, no BFD control packets are sent or received. The adjacency creation takes places once you have configured BFD support for the applicable routing protocols. The first two tasks must be configured to implement BFD support for BGP to reduce the BGP convergence time. The third task is an optional task to help monitor or troubleshoot BFD.

- Prerequisites, page 361
- Restrictions, page 361
- Configuring BFD Session Parameters on the Interface, page 362
- Configuring BFD Support for BGP, page 363

Prerequisites

- Cisco Express Forwarding (CEF) and IP routing must be enabled on all participating routers.
- BGP must be configured on the routers before BFD is deployed. You should implement fast convergence for the routing protocol that you are using. See the IP routing documentation for your version of Cisco IOS XE software for information on configuring fast convergence.

Restrictions

- For the Cisco implementation of BFD support for BGP in Cisco IOS XE Release 2.1, BFD is supported only for IPv4 networks, and only asynchronous mode is supported. In asynchronous mode, either BFD peer can initiate a BFD session.
- BFD works only for directly-connected neighbors. BFD neighbors must be no more than one IP hop away. Multi-hop configurations are not supported.
- Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see the BFD for BGP, page 334.
## Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

### SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number
4. bfd interval milliseconds min_rx milliseconds multiplier interval-multiplier
5. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:** | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | Router# configure terminal |
| **Step 3** interface type number | Enters interface configuration mode. |
| **Example:** | Router(config)# interface FastEthernet 6/0 |
| **Step 4** bfd interval milliseconds min_rx milliseconds multiplier interval-multiplier | Enables BFD on the interface. |
| **Example:** | Router(config-if)# bfd interval 50 min_rx 50 multiplier 5 |
| **Step 5** end | Exits interface configuration mode. |
| **Example:** | Router(config-if)# end |
Configuring BFD Support for BGP

Perform this task to configure BFD support for BGP, so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

- BGP must be running on all participating routers.
- The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See "Configuring BFD Session Parameters on the Interface" for more information.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address fall-over bfd
5. end
6. show bfd neighbors [details]
7. show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics | received prefix-filter | policy[detail]]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 router bgp autonomous-system-number</td>
<td>Specifies a BGP process and enters router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp tag1</td>
<td></td>
</tr>
<tr>
<td>Step 4 neighbor ip-address fall-over bfd</td>
<td>Enables BFD support for failover.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 172.16.10.2 fall-over bfd</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Returns the router to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router)# end
```

<table>
<thead>
<tr>
<th><strong>Step 6</strong> show bfd neighbors [details]</th>
<th>Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.</th>
</tr>
</thead>
</table>

**Example:**

```
Router# show bfd neighbors detail
```

| **Step 7** show ip bgp neighbors [ip-address [received-routes | routes | advertised-routes | paths [regexp] | dampened-routes | flap-statistics] | received prefix-filter| policy[detail]]] | Displays information about BGP and TCP connections to neighbors. |
|-------------------------------------------------------------|-------------------------------------------------|

**Example:**

```
Router# show ip bgp neighbors
```

- What to Do Next, page 364

**What to Do Next**

For more details about BFD, see the "Bidirectional Forwarding Detection" chapter of the Cisco IOS XE IP Routing: BFD Configuration Guide, Release 2.3.

**Enabling BGP MIB Support**

SNMP notifications can be configured on the router and GET operations can be performed from an external management station only after BGP SNMP support is enabled. Perform this task on a router to configure SNMP notifications for the BGP MIB.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. snmp-server enable traps bgp [[state-changes [all] [backward-trans] [limited]] | [threshold prefix]]
4. exit
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> snmp-server enable traps bgp [state-changes [all] [backward-trans] [limited]] [threshold prefix]</td>
<td>Enables BGP support for SNMP operations. Entering this command with no keywords or arguments enables support for all BGP events.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# snmp-server enable traps bgp</td>
<td>• The <code>state-changes</code> keyword is used to enable support for FSM transition events.</td>
</tr>
<tr>
<td></td>
<td>• The <code>all</code> keyword enables support for FSM transitions events.</td>
</tr>
<tr>
<td></td>
<td>• The <code>backward-trans</code> keyword enables support only for backward transition state change events.</td>
</tr>
<tr>
<td></td>
<td>• The <code>limited</code> keyword enables support for backward transition state changes and established state events.</td>
</tr>
<tr>
<td></td>
<td>• The <code>threshold</code> and <code>prefix</code> keywords are used to enable notifications when the configured maximum prefix limit is reached on the specified peer.</td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode, and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Configuration Examples for Configuring Advanced BGP Features**

- Enabling and Disabling BGP Next-Hop Address Tracking Example, page 366
- Adjusting the Delay Interval for BGP Next-Hop Address Tracking Example, page 366
- Configuring BGP Selective Next-Hop Route Filtering Examples, page 366
- Enabling BGP Global NSF Awareness Using Graceful Restart Example, page 366
- Enabling and Disabling BGP Graceful Restart per Neighbor Examples, page 367
- Configuring BGP Route Dampening Example, page 368
- Enabling BGP MIB Support Examples, page 368
Enabling and Disabling BGP Next-Hop Address Tracking Example

In the following example, next-hop address tracking is disabled under the IPv4 address family session:

```bgp
router bgp 50000
    address-family ipv4 unicast
    no bgp nexthop trigger enable
```

Adjusting the Delay Interval for BGP Next-Hop Address Tracking Example

In the following example, the delay interval for next-hop tracking is configured to occur every 20 seconds under the IPv4 address family session:

```bgp
router bgp 50000
    address-family ipv4 unicast
    bgp nexthop trigger delay 20
```

Configuring BGP Selective Next-Hop Route Filtering Examples

The following example shows how to configure BGP selective next-hop route filtering to avoid using a BGP prefix as the next-hop route. If the most specific route that covers the next hop is a BGP route, then the BGP route will be marked as unreachable. The next hop must be an IGP or static route.

```bgp
router bgp 45000
    address-family ipv4 unicast
    bgp nexthop route-map CHECK-BGP
    exit
route-map CHECK-BGP deny 10
    match source-protocol bgp 1
    exit
route-map CHECK-BGP permit 20
    end
```

The following example shows how to configure BGP selective next-hop route filtering to avoid using a BGP prefix as the next-hop route and to ensure that the prefix is more specific than /25.

```bgp
router bgp 45000
    address-family ipv4 unicast
    bgp nexthop route-map CHECK-BGP25
    exit
exit
ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25
route-map CHECK-BGP25 deny 10
    match ip address prefix-list FILTER25
    exit
route-map CHECK-BGP25 deny 20
    match source-protocol bgp 1
    exit
route-map CHECK-BGP25 permit 30
    end
```

Enabling BGP Global NSF Awareness Using Graceful Restart Example

The following example enables BGP NSF awareness globally on all BGP neighbors. The restart time is set to 130 seconds and the stale path time is set to 350 seconds. The configuration of these timers is optional and the preconfigured default values are optimal for most network deployments.

```bgp
configure terminal
router bgp 45000
```
Enabling and Disabling BGP Graceful Restart per Neighbor Examples

The ability to enable or disable the BGP graceful restart capability for an individual BGP neighbor, peer group, or peer session template was introduced. The following example is configured on Router B in the figure below and enables the BGP graceful restart capability for the BGP peer session template named S1 and disables the BGP graceful restart capability for the BGP peer session template named S2. The external BGP neighbor at Router A in the figure below (192.168.1.2) inherits peer session template S1, and the BGP graceful restart capability is enabled for this neighbor. Another external BGP neighbor at Router E in the figure below (192.168.3.2) is configured with the BGP graceful restart capability disabled after inheriting peer session template S2.

Figure 33  Network Topology Showing BGP Neighbors for BGP Graceful Restart

The BGP graceful restart capability is enabled for an individual internal BGP neighbor, 172.21.1.2 at Router C in the figure above, whereas the BGP graceful restart is disabled for the BGP neighbor 172.16.1.2 at Router D in the figure above because it is a member of the peer group PG1. The disabling of BGP graceful restart is configured for all members of the peer group, PG1. The restart and stale-path timers are modified and the BGP sessions are reset.

```
router bgp 45000
  template peer-session S1
    remote-as 40000
    ha-mode graceful-restart
    exit-peer-session
  template peer-session S2
    remote-as 50000
    ha-mode graceful-restart disable
    exit-peer-session
  bgp log-neighbor-changes
  bgp graceful-restart restart-time 150
  bgp graceful-restart stalepath-time 400
  address-family ipv4 unicast
  neighbor PG1 peer-group
```

IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2
neighbor PG1 remote-as 45000
neighbor PG1 ha-mode graceful-restart disable
neighbor 172.16.1.2 peer-group PG1
neighbor 172.21.1.2 remote-as 45000
neighbor 172.21.1.2 activate
neighbor 172.21.1.2 ha-mode graceful-restart
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.1.2 inherit peer-session S1
neighbor 192.168.3.2 remote-as 50000
neighbor 192.168.3.2 inherit peer-session S2
end

clear ip bgp *

To demonstrate how the last configuration instance of the BGP graceful restart capability is applied, the following example initially enables the BGP graceful restart capability globally for all BGP neighbors. A BGP peer group, PG2, is configured with the BGP graceful restart capability disabled. An individual external BGP neighbor, 192.168.1.2 at Router A in the figure above, is then configured to be a member of the peer group, PG2. The last graceful restart configuration instance is applied, and, in this case, the neighbor, 192.168.1.2, inherits the configuration instance from the peer group PG2 and the BGP graceful restart capability is disabled for this neighbor.

```
router bgp 45000
  bgp log-neighbor-changes
  bgp graceful-restart
  address-family ipv4 unicast
  neighbor PG2 peer-group
  neighbor PG2 remote-as 40000
  neighbor PG2 ha-mode graceful-restart disable
  neighbor 192.168.1.2 peer-group PG2
end
clear ip bgp *
```

**Configuring BGP Route Dampening Example**

The following example configures BGP dampening to be applied to prefixes filtered through the route-map named ACCOUNTING:

```
ip prefix-list FINANCE permit 10.0.0.0/8
route-map ACCOUNTING
  match ip address ip prefix-list FINANCE
  set dampening 15 750 2000 60
exit
router bgp 50000
  address-family ipv4
  bgp dampening route-map ACCOUNTING
end
```

**Enabling BGP MIB Support Examples**

The following example enables SNMP support for all supported BGP events:

```
Router(config)# snmp-server enable traps bgp
```

The following verification example shows that SNMP support for BGP is enabled and shown the running-config file:

```
Router# show run | include snmp-server
snmp-server enable traps bgp
```
Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" chapter of the BGP section of the Cisco IOS IP Routing Protocols Configuration Guide.
- If you want to configure BGP neighbor session options, see the "Configuring BGP Neighbor Session Options" module.

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Overview of Cisco BGP conceptual information with links to all the individual BGP modules</td>
<td>&quot;Cisco BGP Overview&quot; module of the Cisco IOS IP Routing Protocols Configuration Guide.</td>
</tr>
<tr>
<td>Information about SNMP and SNMP operations.</td>
<td>&quot;Configuring SNMP Support&quot; section of the Cisco IOS Network Management Configuration Guide.</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT SAFI</td>
<td>MDT SAFI</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-BGP4-MIB</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>
RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1657</td>
<td>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2</td>
</tr>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
<tr>
<td>RFC 1772</td>
<td>Application of the Border Gateway Protocol in the Internet</td>
</tr>
<tr>
<td>RFC 1773</td>
<td>Experience with the BGP Protocol</td>
</tr>
<tr>
<td>RFC 1774</td>
<td>BGP-4 Protocol Analysis</td>
</tr>
<tr>
<td>RFC 1930</td>
<td>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</td>
</tr>
<tr>
<td>RFC 2519</td>
<td>A Framework for Inter-Domain Route Aggregation</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
<tr>
<td>RFC 3392</td>
<td>Capabilities Advertisement with BGP-4</td>
</tr>
<tr>
<td>RFC 4724</td>
<td>Graceful Restart Mechanism for BGP</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for Configuring Advanced BGP Features

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

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

Table 20  Feature Information for Configuring Advanced BGP Features

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Convergence Optimization</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>BGP Graceful Restart per Neighbor</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Graceful Restart per Neighbor feature enables or disables the BGP graceful restart capability for an individual BGP neighbor, including using peer session templates and BGP peer groups. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature:  <code>ha-mode graceful-restart</code>,  <code>neighbor ha-mode graceful-restart</code>,  <code>show ip bgp neighbors</code>.</td>
</tr>
<tr>
<td>BGP MIB Support Enhancements</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. This feature was introduced on the Cisco ASR 1000 Series Routers. The following command was introduced in this feature:  <code>snmp-server enable traps bgp</code>.</td>
</tr>
</tbody>
</table>
Nonstop Forwarding (NSF) awareness allows a router to assist NSF-capable neighbors to continue forwarding packets during a Stateful Switchover (SSO) operation. The BGP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running BGP to forward packets along routes that are already known for a router that is performing an SSO operation. This capability allows the BGP peers of the failing router to retain the routing information that is advertised by the failing router and continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.

This feature was introduced on the Cisco ASR 1000 Series Routers.

The following commands were introduced or modified by this feature: `bgp graceful-restart`, `show ip bgp`, `show ip bgp neighbors`.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Nonstop Forwarding (NSF)</td>
<td>Cisco IOS XE Release 2.1</td>
<td>Nonstop Forwarding (NSF) awareness allows a router to assist NSF-capable neighbors to continue forwarding packets during a Stateful Switchover (SSO) operation. The BGP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running BGP to forward packets along routes that are already known for a router that is performing an SSO operation. This capability allows the BGP peers of the failing router to retain the routing information that is advertised by the failing router and continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified by this feature: <code>bgp graceful-restart</code>, <code>show ip bgp</code>, <code>show ip bgp neighbors</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Selective Address Tracking</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Selective Address Tracking feature introduces the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were modified by this feature: <code>bgp nexthop</code>, <code>neighbor fall-over</code>.</td>
</tr>
</tbody>
</table>
Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster reconvergence time.

This feature was introduced on the Cisco ASR 1000 Series Routers.

The following commands were introduced or modified by this feature: `bfd`, `neighbor fall-over`, `show bfd neighbors`, `show ip bgp neighbors`.
The BGP Support for Next-Hop Address Tracking feature is enabled by default when a supporting Cisco IOS XE software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a bestpath calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

This feature was introduced on the Cisco ASR 1000 Series Routers.

The following command was introduced in this feature: `bgp nexthop`.
BGP Link Bandwidth

The Border Gateway Protocol (BGP) Link Bandwidth feature is used to advertise the bandwidth of an autonomous system exit link as an extended community. This feature is configured for links between directly connected external BGP (eBGP) neighbors. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled. This feature is used with BGP multipath features to configure load balancing over links with unequal bandwidth.

- Finding Feature Information, page 377
- Prerequisites for BGP Link Bandwidth, page 377
- Restrictions for BGP Link Bandwidth, page 378
- Information About BGP Link Bandwidth, page 378
- How to Configure BGP Link Bandwidth, page 379
- Configuration Examples for BGP Link Bandwidth, page 381
- Additional References, page 385
- Feature Information for BGP Link Bandwidth, page 386

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP Link Bandwidth

- BGP load balancing or multipath load balancing must be configured before BGP Link Bandwidth feature is enabled.
- BGP extended community exchange must be enabled between iBGP neighbors to which the link bandwidth attribute is to be advertised.
- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.
Restrictions for BGP Link Bandwidth

- The BGP Link Bandwidth feature can be configured only under IPv4 and VPNv4 address family sessions.
- BGP can originate the link bandwidth community only for directly connected links to eBGP neighbors.
- Both iBGP and eBGP load balancing are supported in IPv4 and VPNv4 address families. However, eiBGP load balancing is supported only in VPNv4 address families.

Information About BGP Link Bandwidth

- BGP Link Bandwidth Overview, page 378
- Link Bandwidth Extended Community Attribute, page 378
- Benefits of the BGP Link Bandwidth Feature, page 378

BGP Link Bandwidth Overview

The BGP Link Bandwidth feature is used to enable multipath load balancing for external links with unequal bandwidth capacity. This feature is enabled under an IPv4 or VPNv4 address family session by entering the `bgp dmzlink-bw` command. This feature supports iBGP, eBGP multipath load balancing, and eiBGP multipath load balancing in Multiprotocol Label Switching (MPLS) VPNs. When this feature is enabled, routes learned from directly connected external neighbor are propagated through the internal BGP (iBGP) network with the bandwidth of the source external link.

The link bandwidth extended community indicates the preference of an autonomous system exit link in terms of bandwidth. This extended community is applied to external links between directly connected eBGP peers by entering the `neighbor dmzlink-bw` command. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled with the `neighbor send-community` command.

Link Bandwidth Extended Community Attribute

The link bandwidth extended community attribute is a 4-byte value that is configured for a link on the demilitarized zone (DMZ) interface that connects two single hop eBGP peers. The link bandwidth extended community attribute is used as a traffic sharing value relative to other paths while traffic is being forwarded. Two paths are designated as equal for load balancing if the weight, local-pref, as-path length, Multi Exit Discriminator (MED), and Interior Gateway Protocol (IGP) costs are the same.

Benefits of the BGP Link Bandwidth Feature

The BGP Link Bandwidth feature allows BGP to be configured to send traffic over multiple iBGP or eBGP learned paths where the traffic that is sent is proportional to the bandwidth of the links that are used to exit the autonomous system. The configuration of this feature can be used with eBGP and iBGP multipath features to enable unequal cost load balancing over multiple links. Unequal cost load balancing over links with unequal bandwidth was not possible in BGP before the BGP Link Bandwidth feature was introduced.
How to Configure BGP Link Bandwidth

- Configuring BGP Link Bandwidth, page 379
- Verifying BGP Link Bandwidth Configuration, page 380

Configuring BGP Link Bandwidth

To configure the BGP Link Bandwidth feature, perform the steps in this section.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]`
5. `bgp dmzlink-bw`
6. `neighbor ip-address dmzlink-bw`
7. `neighbor ip-address send-community [both | extended | standard]`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables higher privilege levels, such as privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router# enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 50000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4** address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]

Example:
```
Router(config-router)# address-family ipv4
```

**Purpose**

Enters address family configuration mode.

- The BGP Link Bandwidth feature is supported only under the IPv4 and VPNv4 address families.

**Step 5** bgp dmzlink-bw

Example:
```
Router(config-router-af)# bgp dmzlink-bw
```

**Purpose**

Configures BGP to distribute traffic proportionally to the bandwidth of the link.

- This command must be entered on each router that contains an external interface that is to be used for multipath load balancing.

**Step 6** neighbor ip-address dmzlink-bw

Example:
```
Router(config-router-af)# neighbor 172.16.1.1 dmzlink-bw
```

**Purpose**

Configures BGP to include the link bandwidth attribute for routes learned from the external interface specified IP address.

- This command must be configured for each eBGP link that is to be configured as a multipath. Enabling this command allows the bandwidth of the external link to be propagated through the link bandwidth extended community.

**Step 7** neighbor ip-address send-community [both | extended | standard]

Example:
```
Router(config-router-af)# neighbor 10.10.10.1 send-community extended
```

**Purpose**

(Optional) Enables community and/or extended community exchange with the specified neighbor.

- This command must be configured for iBGP peers to which the link bandwidth extended community attribute is to be propagated.

**Step 8** end

Example:
```
Router(config-router-af)# end
```

**Purpose**

Exits address family configuration mode, and enters Privileged EXEC mode.

---

### Verifying BGP Link Bandwidth Configuration

To verify the BGP Link Bandwidth feature, perform the steps in this section.

**SUMMARY STEPS**

1. `enable`
2. `show ip bgp ip-address [longer-prefixes [injected] | shorter-prefixes [mask-length]]`
3. `show ip route [[ip-address [mask] [longer-prefixes]] | [protocol [process-id]] | [list access-list-number | access-list-name] | [static download]]`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables higher privilege levels, such as privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** show ip bgp ip-address [longer-prefixes | Displays information about the TCP and BGP connections to |
| injected] | neighbors. |
| shorter-prefixes [mask-length]] | • The output displays the status of the link bandwidth |
| Example:          | configuration. The bandwidth of the link is shown in kilobytes. |
| Router# show ip bgp 10.0.0.0 |         |

| **Step 3** show ip route [ip-address [mask] [longer- | Displays the current state of the routing table. |
| prefixes]] [protocol [process-id]] [list access- | • The output displays traffic share values, including the weights of |
| list-number | access-list-name] [static | the links that are used to direct traffic proportionally to the |
| download]] | bandwidth of each link. |
| Example:          |         |
| Router# show ip route 10.0.0.0 |         |
systems connected by three links that each carry a different amount of bandwidth (unequal cost links). Multipath load balancing is enabled and traffic is balanced proportionally.

**Figure 34  BGP Link Bandwidth Configuration**

 Autonomous system 100

 Router A

 10.10.10.1/24

 Router B

 10.10.2/24

 Router C

 10.10.3/24

 Bandwidth in bits per second

 500

 1733

 5000

 Autonomous system 200

 Router D

 172.16.1.1/24

 Router E

 172.16.2.2/24

**Router A Configuration**

In the following example, Router A is configured to support iBGP multipath load balancing and to exchange the BGP extended community attribute with iBGP neighbors:

Router A(config)# router bgp 100
Router A(config-router)# neighbor 10.10.10.2 remote-as 100
Router A(config-router)# neighbor 10.10.10.2 update-source Loopback 0
Router A(config-router)# neighbor 10.10.10.3 remote-as 100
Router A(config-router)# neighbor 10.10.10.3 update-source Loopback 0
Router A(config-router)# address-family ipv4
Router A(config-router)# bgp dmzlink-bw
Router A(config-router-af)# neighbor 10.10.10.2 activate
Router A(config-router-af)# neighbor 10.10.10.2 send-community both
Router A(config-router-af)# neighbor 10.10.10.3 activate
Router A(config-router-af)# neighbor 10.10.10.3 send-community both
Router A(config-router-af)# maximum-paths ibgp 6
Router B Configuration

In the following example, Router B is configured to support multipath load balancing, to distribute Router D and Router E link traffic proportionally to the bandwidth of each link, and to advertise the bandwidth of these links to iBGP neighbors as an extended community:

Router B(config)# router bgp 100
Router B(config-router)# neighbor 10.10.10.1 remote-as 100
Router B(config-router)# neighbor 10.10.10.1 update-source Loopback 0
Router B(config-router)# neighbor 10.10.10.3 remote-as 100
Router B(config-router)# neighbor 10.10.10.3 update-source Loopback 0
Router B(config-router)# neighbor 172.16.1.1 remote-as 200
Router B(config-router)# neighbor 172.16.1.1 ebgp-multihop 1
Router B(config-router)# neighbor 172.16.2.2 remote-as 200
Router B(config-router)# neighbor 172.16.2.2 ebgp-multihop 1
Router B(config-router)# address-family ipv4
Router B(config-router-af)# bgp dmzlink-bw
Router B(config-router-af)# neighbor 10.10.10.1 activate
Router B(config-router-af)# neighbor 10.10.10.1 next-hop-self
Router B(config-router-af)# neighbor 10.10.10.1 send-community both
Router B(config-router-af)# neighbor 10.10.10.2 activate
Router B(config-router-af)# neighbor 10.10.10.2 next-hop-self
Router B(config-router-af)# neighbor 10.10.10.2 send-community both
Router B(config-router-af)# neighbor 172.16.1.1 activate
Router B(config-router-af)# neighbor 172.16.1.1 dmzlink-bw
Router B(config-router-af)# neighbor 172.16.2.2 activate
Router B(config-router-af)# neighbor 172.16.2.2 dmzlink-bw
Router B(config-router-af)# maximum-paths ibgp 6
Router B(config-router-af)# maximum-paths 6

Router C Configuration

In the following example, Router C is configured to support multipath load balancing and to advertise the bandwidth of the link with Router E to iBGP neighbors as an extended community:

Router C(config)# router bgp 100
Router C(config-router)# neighbor 10.10.10.1 remote-as 100
Router C(config-router)# neighbor 10.10.10.1 update-source Loopback 0
Router C(config-router)# neighbor 10.10.10.2 remote-as 100
Router C(config-router)# neighbor 10.10.10.2 update-source Loopback 0
Router C(config-router)# neighbor 172.16.3.30 remote-as 200
Router C(config-router)# neighbor 172.16.3.30 ebgp-multihop 1
Router C(config-router)# address-family ipv4
Router C(config-router-af)# bgp dmzlink-bw
Router C(config-router-af)# neighbor 10.10.10.1 activate
Router C(config-router-af)# neighbor 10.10.10.1 send-community both
Router C(config-router-af)# neighbor 10.10.10.1 next-hop-self
Router C(config-router-af)# neighbor 10.10.10.2 activate
Router C(config-router-af)# neighbor 10.10.10.2 send-community both
Router C(config-router-af)# neighbor 10.10.10.2 next-hop-self
Router C(config-router-af)# neighbor 172.16.3.3 activate
Router C(config-router-af)# neighbor 172.16.3.3 dmzlink-bw
Verifying BGP Link Bandwidth

The examples in this section show the verification of this feature on Router A and Router B.

Router B

In the following example, the `show ip bgp` command is entered on Router B to verify that two unequal cost best paths have been installed into the BGP routing table. The bandwidth for each link is displayed with each route.

```plaintext
Router B# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 48
Paths: (2 available, best #2)
Multipath: eBGP
  Advertised to update-groups:
    1          2
  200
    172.16.1.1 from 172.16.1.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 278 kbytes
  200
    172.16.2.2 from 172.16.2.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 625 kbytes
```

Router A

In the following example, the `show ip bgp` command is entered on Router A to verify that the link bandwidth extended community has been propagated through the iBGP network to Router A. The output shows that a route for each exit link (on Router B and Router C) to autonomous system 200 has been installed as a best path in the BGP routing table.

```plaintext
Router A# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 48
Paths: (3 available, best #3)
Multipath: eBGP
  Advertised to update-groups:
    1          2
  200
    172.16.1.1 from 172.16.1.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath
      Extended Community: 0x0:0:0
      DMZ-Link Bw 278 kbytes
  200
    172.16.2.2 from 172.16.2.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 625 kbytes
  200
    172.16.3.3 from 172.16.3.3 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 2500 kbytes
```
Router A

In the following example, the `show ip route` command is entered on Router A to verify the multipath routes that are advertised and the associated traffic share values:

```
Router A# show ip route 192.168.1.0
Routing entry for 192.168.1.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 200, type internal
  Last update from 172.168.1.1 00:01:43 ago
  Routing Descriptor Blocks:
    * 172.168.1.1, from 172.168.1.1, 00:01:43 ago
       Route metric is 0, traffic share count is 13
       AS Hops 1, BGP network version 0
       Route tag 200
    172.168.2.2, from 172.168.2.2, 00:01:43 ago
       Route metric is 0, traffic share count is 30
       AS Hops 1, BGP network version 0
       Route tag 200
    172.168.3.3, from 172.168.3.3, 00:01:43 ago
       Route metric is 0, traffic share count is 120
       AS Hops 1, BGP network version 0
       Route tag 200
```

Additional References

The following sections provide references related to the BGP Link Bandwidth feature.

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
</tbody>
</table>
| BGP multipath load sharing for both eBGP and iBGP in an MPLS-VPN | "BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN"
| iBGP multipath load sharing | "iBGP Multipath Load Sharing"
| Cisco IOS master command list, all releases | *Cisco IOS Master Command List, All Releases* |

**Standards**

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for BGP Link Bandwidth

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 21 Feature Information for BGP Link Bandwidth

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Link Bandwidth</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were added or modified by this feature: bgp dmzlink-bw, neighbor dmzlink-bw.</td>
</tr>
</tbody>
</table>
BGP Link Bandwidth

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
iBGP Multipath Load Sharing

This feature module describes the iBGP Multipath Load Sharing feature. This feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router.

- Finding Feature Information, page 389
- iBGP Multipath Load Sharing Overview, page 389
- How To Configure iBGP Multipath Load Sharing, page 392
- Configuration Examples, page 395
- Additional References, page 396
- Feature Information for iBGP Multipath Load Sharing, page 397

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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.

iBGP Multipath Load Sharing Overview

When a Border Gateway Protocol (BGP) speaking router with no local policy configured receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router will choose one iBGP path as the best path. The best path is then installed in the IP routing table of the router. For example, in the figure below, although there are three paths to autonomous system 200,
Router 2 determines that one of the paths to autonomous system 200 is the best path and uses this path only to reach autonomous system 200.

**Figure 35  Non-MPLS Topology with One Best Path**

![Non-MPLS Topology with One Best Path](image1)

The iBGP Multipath Load Sharing feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router. For example, on router 2 in the figure below, the paths to routers 3, 4, and 5 are configured as multipaths and can be used to reach autonomous system 200, thereby equally sharing the load to autonomous system 200.

**Figure 36  Non-MPLS Topology with Three Multipaths**

![Non-MPLS Topology with Three Multipaths](image2)

The iBGP Multipath Load Sharing feature functions similarly in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) with a service provider backbone. For example, on router PE1 in the figure
below, the paths to routers PE2, PE3, and PE4 can be selected as multipaths and can be used to equally share the load to site 2.

**Figure 37** **MPLS VPN with Three Multipaths**

For multiple paths to the same destination to be considered as multipaths, the following criteria must be met:

- All attributes must be the same. The attributes include weight, local preference, autonomous system path (entire attribute and not just length), origin code, Multi Exit Discriminator (MED), and Interior Gateway Protocol (IGP) distance.
- The next hop router for each multipath must be different.

Even if the criteria are met and multiple paths are considered multipaths, the BGP speaking router will still designate one of the multipaths as the best path and advertise this best path to its neighbors.

- **Benefits of iBGP Multipath Load Sharing**, page 391
- **Restrictions on iBGP Multipath Load Sharing**, page 391

**Benefits of iBGP Multipath Load Sharing**

Configuring multiple iBGP best paths enables a router to evenly share the traffic destined for a particular site.

**Restrictions on iBGP Multipath Load Sharing**

**Route Reflector Limitation**

With multiple iBGP paths installed in a routing table, a route reflector will advertise only one of the paths (one next hop).

**Memory Consumption Restriction**

Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses approximately 350 bytes of additional memory. We recommend not using this feature on a router with a low amount of available memory and especially when the router is carrying a full Internet routing table.
How To Configure iBGP Multipath Load Sharing

• Configuring iBGP Multipath Load Sharing, page 392
• Verifying iBGP Multipath Load Sharing, page 392
• Monitoring and Maintaining iBGP Multipath Load Sharing, page 394

Configuring iBGP Multipath Load Sharing

To configure the iBGP Multipath Load Sharing feature, use the following command in router configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-router)# maximum-paths ibgp maximum-number</td>
<td>Controls the maximum number of parallel iBGP routes that can be installed in a routing table.</td>
</tr>
</tbody>
</table>

Verifying iBGP Multipath Load Sharing

To verify that the iBGP Multipath Load Sharing feature is configured correctly, perform the following steps:

**SUMMARY STEPS**

1. Enter the `show ip bgp network-number` EXEC command to display attributes for a network in a non-MPLS topology, or the `show ip bgp vpv4 all ip-prefix` EXEC command to display attributes for a network in an MPLS VPN:

Example:

```
Router# show ip bgp 10.22.22.0
BGP routing table entry for 10.22.22.0/24, version 119
Paths:(6 available, best #1)
Multipath:iBGP
```

2. In the display resulting from the `show ip bgp network-number` EXEC command or the `show ip bgp vpv4 all ip-prefix` EXEC command, verify that the intended multipaths are marked as "multipaths." Notice that one of the multipaths is marked as "best."

3. Enter the `show ip route ip-address` EXEC command to display routing information for a network in a non-MPLS topology or the `show ip route vrf vrf-name ip-prefix` EXEC command to display routing information for a network in an MPLS VPN:

4. Verify that the paths marked as "multipath" in the display resulting from the `show ip bgp ip-prefix` EXEC command or the `show ip bgp vpv4 all ip-prefix` EXEC command are included in the routing information. (The routing information is displayed after performing Step 3.)

**DETAILED STEPS**

**Step 1**

Enter the `show ip bgp network-number` EXEC command to display attributes for a network in a non-MPLS topology, or the `show ip bgp vpv4 all ip-prefix` EXEC command to display attributes for a network in an MPLS VPN:
Step 2  In the display resulting from the `show ip bgp network-number` EXEC command or the `show ip bgp vpnv4 all ip-prefix` EXEC command, verify that the intended multipaths are marked as "multipaths." Notice that one of the multipaths is marked as "best."

Step 3  Enter the `show ip route ip-address` EXEC command to display routing information for a network in a non-MPLS topology or the `show ip route vrf vrf-name ip-prefix` EXEC command to display routing information for a network in an MPLS VPN:

Example:

```
Router# show ip route 10.22.22.0
Routing entry for 10.22.22.0/24
Known via "bgp 1", distance 200, metric 0
Tag 22, type internal
```
Step 4 Verify that the paths marked as "multipath" in the display resulting from the `show ip bgp ip-prefix EXEC` command or the `show ip bgp vpnv4 all ip-prefix EXEC` command are included in the routing information. (The routing information is displayed after performing Step 3.)

## Monitoring and Maintaining iBGP Multipath Load Sharing

To display iBGP Multipath Load Sharing information, use the following commands in EXEC mode, as needed:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router# show ip bgp ip-prefix</code></td>
<td>Displays attributes and multipaths for a network in a non-MPLS topology.</td>
</tr>
<tr>
<td><code>Router# show ip bgp vpnv4 all ip-prefix</code></td>
<td>Displays attributes and multipaths for a network in an MPLS VPN.</td>
</tr>
<tr>
<td><code>Router# show ip route ip-prefix</code></td>
<td>Displays routing information for a network in a non-MPLS topology.</td>
</tr>
</tbody>
</table>
### Configuration Examples

Both examples assume that the appropriate attributes for each path are equal and that the next hop router for each multipath is different.

- Example iBGP Multipath Load Sharing in a Non-MPLS Topology, page 395
- Example iBGP Multipath Load Sharing in an MPLS VPN Topology, page 396

### Example iBGP Multipath Load Sharing in a Non-MPLS Topology

The following example shows how to set up the iBGP Multipath Load Sharing feature in a non-MPLS topology (see the figure below).

![Non-MPLS Topology Example](image)

**Figure 38** Non-MPLS Topology Example

**Router 2 Configuration**

```
routerr bgp 100
maximum-paths ibgp 3
```
Example iBGP Multipath Load Sharing in an MPLS VPN Topology

The following example shows how to set up the iBGP Multipath Load Sharing feature in an MPLS VPN topology (see the figure below).

**Figure 39  MPLS VPN Topology Example**

Router PE1 Configuration

```
router bgp 100
address-family ipv4 unicast vrf site2
maximum-paths ibgp 3
```

Addtional References

The following sections provide references related to the iBGP Multipath Load Sharing feature.

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP multipath load sharing for both eBGP and iBGP in an MPLS-VPN</td>
<td>&quot;BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN&quot;</td>
</tr>
<tr>
<td>Advertising the bandwidth of an autonomous system exit link as an extended community</td>
<td>&quot;BGP Link Bandwidth&quot;</td>
</tr>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
</tbody>
</table>
Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for iBGP Multipath Load Sharing

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 22  Feature Information for iBGP Multipath Load Sharing**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>iBGP multipath load sharing</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were modified by this feature: maximum paths ibgp, show ip bgp, show ip bgp vpnv4, show ip route, show ip route vrf.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for both eBGP and iBGP in an MPLS-VPN feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multihomed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

- Finding Feature Information, page 399
- Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 400
- Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 400
- Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 400
- How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 402
- Configuration Examples for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 405
- Where to Go Next, page 406
- Additional References, page 406
- Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 408

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
**Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

*Load Balancing is Configured Under CEF*
Cisco Express Forwarding (CEF) or distributed CEF (dCEF) must be enabled on all participating routers.

**Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

*Address Family Support*
This feature is configured on a per VPN routing and forwarding instance (VRF) basis. This feature can be configured under only the IPv4 VRF address family.

*Memory Consumption Restriction*
Each BGP multipath routing table entry will use additional memory. We recommend that you do not use this feature on a router with a low amount of available memory and especially if router is carries full Internet routing tables.

*Route Reflector Limitation*
When multiple iBGP paths installed in a routing table, a route reflector will advertise only one paths (next hop). If a router is behind a route reflector, all routers that are connected to multihomed sites will not be advertised unless a different route distinguisher is configured for each VRF.

**Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

- Multipath Load Sharing Between eBGP and iBGP, page 400
- eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network, page 401
- eBGP and iBGP Multipath Load Sharing With Route Reflectors, page 402
- Benefits of Multipath Load Sharing for Both eBGP and iBGP, page 402

**Multipath Load Sharing Between eBGP and iBGP**

A BGP routing process will install a single path as the best path in the routing information base (RIB) by default. The `maximum-paths` command allows you to configure BGP to install multiple paths in the RIB for multipath load sharing. BGP uses the best path algorithm to still select a single multipath as the best path and advertise the best path to BGP peers.
The number of paths of multipaths that can be configured is documented on the `maximum-paths` command reference page.

Load balancing over the multipaths is performed by CEF. CEF load balancing is configured on a per-packet round robin or on a per session (source and destination pair) basis. For information about CEF, refer to the "Cisco Express Forwarding Overview" documentation:

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature is enabled only under the IPv4 VRF address family configuration mode. When enabled, this feature can perform load balancing on eBGP and/or iBGP paths that are imported into the VRF. The number of multipaths is configured on a per VRF basis. Separate VRF multipath configurations are isolated by unique route distinguisher.

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature operates within the parameters of configured outbound routing policy.

---

**eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network**

The figure below shows a service provider BGP MPLS network that connects two remote networks to PE router 1 and PE router 2. PE router 1 and PE router 2 are both configured for VPNv4 unicast iBGP peering. Network 2 is a multihomed network that is connected to PE router 1 and PE router 2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.

**Figure 40 A Service Provider BGP MPLS Network**

PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF of Network 1. The multipaths will be used by CEF to perform load balancing. IP traffic that is sent from Network 2 to PE router 1 and PE router 2 will be sent across the eBGP paths as IP traffic. IP traffic that is sent across the iBGP path will be sent as MPLS traffic, and MPLS traffic that is sent across an eBGP path will be sent as IP traffic. Any prefix that is advertised from Network 2 will be received by PE router 1 through route distinguisher (RD) 21 and RD 22. The advertisement through RD 21 will be carried in IP
packets, and the advertisement through RD 22 will be carried in MPLS packets. Both paths can be selected as multipaths for VRF1 and installed into the VRF1 RIB.

**eBGP and iBGP Multipath Load Sharing With Route Reflectors**

The figure below shows a topology that contains three PE routers and a route reflector, all configured for iBGP peering. PE router 2 and PE router 3 each advertise an equal preference eBGP path to PE router 1. By default, the route reflector will choose only one path and advertise PE router 1.

*Figure 41 A Topology with a Route Reflector*

For all equal preference paths to PE router 1 to be advertised through the route reflector, you must configure each VRF with a different RD. The prefixes received by the route reflector will be recognized differently and advertised to PE router 1.

**Benefits of Multipath Load Sharing for Both eBGP and iBGP**

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both eBGP and iBGP paths.

**How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

- Configuring Multipath Load Sharing for Both eBGP an iBGP, page 402
- Verifying Multipath Load Sharing for Both eBGP an iBGP, page 404

**Configuring Multipath Load Sharing for Both eBGP an iBGP**

To configure this feature, perform the steps in this section.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 vrf vrf-name
5. maximum-paths eibgp number [import number]
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables higher privilege levels, such as privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 router bgp autonomous-system-number</td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 40000</td>
<td></td>
</tr>
<tr>
<td>Step 4 address-family ipv4 vrf vrf-name</td>
<td>Places the router in address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Separate VRF multipath configurations are isolated by unique route distinguisher.</td>
</tr>
<tr>
<td>Router(config-router)# address-family ipv4 vrf RED</td>
<td></td>
</tr>
<tr>
<td>Step 5 maximum-paths eibgp number [import number]</td>
<td>Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: The maximum-paths eibgp command can be configured only under the IPv4 VRF address family configuration mode and cannot be configured in any other address family configuration mode.</td>
</tr>
<tr>
<td>Router(config-router-af)# maximum-paths eibgp 6</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 6**  
end | Exits address family configuration mode, and enters Privileged EXEC mode.

**Example:**

```
Router(config-router-af)# end
```

---

## Verifying Multipath Load Sharing for Both eBGP an iBGP

To verify this feature, perform the steps in this section

### SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** `[neighbor-address] [advertised-routes] [dampened-routes] [flap-statistics] [paths[regexp]] [received prefix-filter] [received-routes] [routes]]`
3. **show ip bgp vpnv4** `{all | rd route-distinguisher} vrf vrf-name`
4. **show ip route vrf** `vrf-name`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
enable | Enables higher privilege levels, such as privileged EXEC mode.  
• Enter your password if prompted.  

**Example:**

```
Router> enable
```

| **Step 2**  
**show ip bgp neighbors** `[neighbor-address] [advertised-routes] [dampened-routes] [flap-statistics] [paths[regexp]] [received prefix-filter] [received-routes] [routes]]` | Displays information about the TCP and BGP connections to neighbors.  

**Example:**

```
Router# show ip bgp neighbors
```

| **Step 3**  
**show ip bgp vpnv4** `{all | rd route-distinguisher} vrf vrf-name}` | Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP.  

**Example:**

```
Router# show ip bgp vpnv4 vrf RED
```
### Command or Action

**Step 4**  
`show ip route vrf vrf-name`

**Example:**

```
Router# show ip route vrf RED
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show ip route vrf</code></td>
<td>Displays the IP routing table associated with a VRF instance. The show ip route vrf command is used to verify that the VRF is in the routing table.</td>
</tr>
</tbody>
</table>

---

### Configuration Examples for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

- eBGP and iBGP Multipath Load Sharing Configuration Example,  page 405
- eBGP and iBGP Multipath Load Sharing Verification Examples,  page 405

### eBGP and iBGP Multipath Load Sharing Configuration Example

This following configuration example configures a router in address-family mode to select six BGP routes (eBGP or iBGP) as multipaths:

```
Router(config)# router bgp 40000
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# maximum-paths eibgp 6
Router(config-router-af)# end
```

### eBGP and iBGP Multipath Load Sharing Verification Examples

To verify that iBGP and eBGP routes have been configured for load sharing, use the `show ip bgp vpnv4` EXEC command or the `show ip route vrf` EXEC command.

In the following example, the `show ip bgp vpnv4` command is entered to display multipaths installed in the VPNv4 RIB:

```
Router# show ip bgp vpnv4 all 10.22.22.0
BGP routing table entry for 10:1:22.22.22.0/24, version 19
Paths:(5 available, best #5)
Multipath:eiBGP
Advertised to non peer-group peers:
10.0.0.2 10.0.0.3 10.0.0.4 10.0.0.5
22
10.0.0.2 (metric 20) from 10.0.0.4 (10.0.0.4)
 Origin IGP, metric 0, localpref 100, valid, internal, multipath
 Extended Community:0x0:0:0 RT:100:1 0x0:0:0
 Originator:10.0.0.2, Cluster list:10.0.0.4
22
10.0.0.2 (metric 20) from 10.0.0.5 (10.0.0.5)
 Origin IGP, metric 0, localpref 100, valid, internal, multipath
 Extended Community:0x0:0:0 RT:100:1 0x0:0:0
 Originator:10.0.0.2, Cluster list:10.0.0.5
22
10.0.0.2 (metric 20) from 10.0.0.2 (10.0.0.2)
 Origin IGP, metric 0, localpref 100, valid, internal, multipath
 Extended Community:RT:100:1 0x0:0:0
22
10.0.0.2 (metric 20) from 10.0.0.3 (10.0.0.3)
```
In the following example, the **show ip route vrf** command is entered to display multipath routes in the VRF table:

```
Router# show ip route vrf PATH 10.22.22.0
Routing entry for 10.22.22.0/24
Known via "bgp 1", distance 20, metric 0
Tag 22, type external
Last update from 10.1.1.12 01:59:31 ago
Routing Descriptor Blocks:
* 10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.4, 01:59:31 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.5, 01:59:31 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.3, 01:59:31 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
10.1.1.12, from 10.1.1.12, 01:59:31 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
```

### Where to Go Next

For information about advertising the bandwidth of an autonomous system exit link as an extended community, refer to the "BGP Link Bandwidth" document.

### Additional References

For additional information related to BGP Multipath Load sharing for Both eBGP and iBGP in an MPLS VPN, refer to the following references:

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>Comprehensive BGP link bandwidth configuration examples and tasks</td>
<td>&quot;BGP Link Bandwidth&quot; module</td>
</tr>
<tr>
<td>CEF configuration tasks</td>
<td>&quot;Cisco Express Forwarding Overview&quot; module</td>
</tr>
</tbody>
</table>
Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: <a href="http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP4)</td>
</tr>
<tr>
<td>RFC 2547</td>
<td>BGP/MPLS VPNs</td>
</tr>
<tr>
<td>RFC 2858</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
# Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Loadsharing IP Packets over More Than Six Parallel Paths

This document describes the Loadsharing IP Packets over More Than Six Parallel Paths feature, which increases the maximum number of parallel routes that can be installed to the routing table for multipath loadsharing.

- Finding Feature Information, page 409
- Overview of Loadsharing IP Packets over More Than Six Parallel Paths, page 409
- Additional References, page 410
- Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths, page 411

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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.

Overview of Loadsharing IP Packets over More Than Six Parallel Paths

The Loadsharing IP Packets over More Than Six Parallel Paths feature increases the maximum number of parallel routes that can be installed to the routing table. The maximum number has been increased from six to sixteen for the following commands:

- maximum-paths
- maximum-paths eibgp
- maximum-paths ibgp

The output of the show ip route summary command has been updated to display the number of parallel routes supported by the routing table.

The benefits of this feature include the following:

- More flexible configuration of parallel routes in the routing table.
• Ability to configure multipath loadsharing over more links to allow for the configuration of higher-bandwidth aggregation using lower-speed links.

### Additional References

For additional information related to multipath loadsharing and the configuration of parallel routes, see the following references:

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>eBGP Multipath Load Sharing</td>
<td>&quot;BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN&quot; module</td>
</tr>
<tr>
<td>iBGP Multipath Load Sharing</td>
<td>&quot;iBGP Multipath Load Sharing&quot; module</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loadsharing IP Packets over More Than Six Parallel Paths</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified by this feature: <code>maximum-paths</code>, <code>maximum-paths eibgp</code>, <code>maximum-paths ibgp</code>, <code>show ip route summary</code></td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Policy Accounting Output Interface Accounting

Border Gateway Protocol (BGP) policy accounting (PA) measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting was previously available on an input interface only. The BGP Policy Accounting Output Interface Accounting feature introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. Counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

- Finding Feature Information, page 413
- Prerequisites for BGP PA Output Interface Accounting, page 413
- Information About BGP PA Output Interface Accounting, page 413
- How to Configure BGP PA Output Interface Accounting, page 415
- Configuration Examples for BGP PA Output Interface Accounting, page 421
- Additional References, page 422
- Feature Information for BGP Policy Accounting Output Interface Accounting, page 423
- Glossary, page 424

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP PA Output Interface Accounting

Before using the BGP Policy Accounting Output Interface Accounting feature, you must enable BGP and Cisco Express Forwarding or distributed CEF on the router.

Information About BGP PA Output Interface Accounting

- BGP PA Output Interface Accounting, page 414
BGP PA Output Interface Accounting

Policy accounting using BGP measures and classifies IP traffic that is sent to, or received from, different peers. Originally, BGP PA was available on an input interface only. BGP PA output interface accounting introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. Counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

Using the BGP table-map command, prefixes added to the routing table are classified by BGP attribute, autonomous system number, or autonomous system path. Packet and byte counters are incremented per input or output interface. A Cisco policy-based classifier maps the traffic into one of eight possible buckets that represent different traffic classes.

Using BGP PA, you can account for traffic according to its origin or the route it traverses. Service providers (SPs) can identify and account for all traffic by customer and can bill accordingly. In the figure below, BGP PA can be implemented in Router A to measure packet and byte volumes in autonomous system buckets. Customers are billed appropriately for traffic that is routed from a domestic, international, or satellite source.

Figure 42 Sample Topology for BGP Policy Accounting

BGP policy accounting using autonomous system numbers can be used to improve the design of network circuit peering and transit agreements between Internet service providers (ISPs).

Benefits of BGP PA Output Interface Accounting

Accounting for IP Traffic Differentially

BGP policy accounting classifies IP traffic by autonomous system number, autonomous system path, or community list string, and increments packet and byte counters. Policy accounting can also be based on the
source address. Service providers can account for traffic and apply billing according to the origin of the traffic or the route that specific traffic traverses.

**Efficient Network Circuit Peering and Transit Agreement Design**
Implementing BGP policy accounting on an edge router can highlight potential design improvements for peering and transit agreements.

# How to Configure BGP PA Output Interface Accounting

- Specifying the Match Criteria for BGP PA, page 415
- Classifying the IP Traffic and Enabling BGP PA, page 416
- Verifying BGP Policy Accounting, page 419

## Specifying the Match Criteria for BGP PA

The first task in configuring BGP PA is to specify the criteria that must be matched. Community lists, autonomous system paths, or autonomous system numbers are examples of BGP attributes that can be specified and subsequently matched using a route map. Perform this task to specify the BGP attribute to use for BGP PA and to create the match criteria in a route map.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip community-list {standard-list-number|expanded-list-number[regular-expression] | [standard| expanded} community-list-name {permit|deny} {community-number|regular-expression}`
4. `route-map map-name [permit|deny] [sequence-number]`
5. `match community-list community-list-name [exact]`
6. `set traffic-index bucket-number`
7. `exit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 3**  
`ip community-list {standard-list-number|expanded-list-number|regular-expression} | {standard|expanded} community-list-name} {permit|deny} {community-number| regular-expression}

**Example:**
```
Router(config)# ip community-list 30 permit 100:190
```

**Step 4**  
`route-map map-name [permit|deny] [sequence-number]

**Example:**
```
Router(config)# route-map set_bucket permit 10
```

**Step 5**  
`match community-list community-list-number [exact]

**Example:**
```
Router(config-route-map)# match community-list 30
```

**Step 6**  
`set traffic-index bucket-number

**Example:**
```
Router(config-route-map)# set traffic-index 2
```

**Step 7**  
`exit

**Example:**
```
Router(config-route-map)# exit
```

---

### Classifying the IP Traffic and Enabling BGP PA

After a route map has been defined to specify match criteria, you must configure a way to classify the IP traffic before enabling BGP policy accounting.

Using the `table-map` command, BGP classifies each prefix that it adds to the routing table according to the match criteria. When the `bgp-policy accounting` command is configured on an interface, BGP policy accounting is enabled.
Perform this task to classify the IP traffic and enable BGP policy accounting.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp as-number
4. table-map route-map-name
5. network network-number [mask network-mask]
6. neighbor ip-address remote-as as-number
7. exit
8. interface type number
9. ip address ip-address mask
10. bgp-policy accounting [input] [output] [source]
11. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 2</strong> configure terminal</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 3</strong> router bgp as-number</th>
<th>Configures a BGP routing process and enters router configuration mode for the specified routing process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>• The as-number argument identifies a BGP autonomous system number.</td>
</tr>
<tr>
<td>Router(config)# router bgp 65000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 4</strong> table-map route-map-name</th>
<th>Classifies BGP prefixes entered in the routing table.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# table-map set_bucket</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>network network-number [mask network-mask]</th>
<th>Specifies a network to be advertised by the BGP routing process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-router)# network 10.15.1.0 mask 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>neighbor ip-address remote-as as-number</td>
<td>Specifies a BGP peer by adding an entry to the BGP routing table.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-router)# neighbor 10.14.1.1 remote-as 65100</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>exit</td>
<td>Exits router configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>interface type number</td>
<td>Specifies the interface type and number and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface POS 7/0</td>
<td>• The type argument identifies the type of interface. • The number argument identifies the slot and port numbers of the interface. The space between the interface type and number is optional.</td>
</tr>
<tr>
<td>Step 9</td>
<td>ip address ip-address mask</td>
<td>Configures the interface with an IP address.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# ip-address 10.15.1.2 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>bgp-policy accounting [input] [output] [source]</td>
<td>Enables BGP policy accounting for the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# bgp-policy accounting input source</td>
<td>• Use the optional input or output keyword to account for traffic either entering or leaving the router. By default, BGP policy accounting is based on traffic entering the router. • Use the optional source keyword to account for traffic based on source address.</td>
</tr>
<tr>
<td>Step 11</td>
<td>exit</td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Verifying BGP Policy Accounting

Perform this task to verify that BGP policy accounting is operating.

SUMMARY STEPS

1. `show ip cef [network[mask]] [detail]`
2. `show ip bgp [network] [network-mask] [longer-prefixes]`
3. `show cef interface [type number] policy-statistics [input] [output]`
4. `show cef interface [type number] [statistics] [detail]`

DETAILED STEPS

Step 1

`show ip cef [network[mask]] [detail]`

Enter the `show ip cef` command with the `detail` keyword to learn which accounting bucket is assigned to a specified prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that accounting bucket number 4 (traffic_index 4) is assigned to this prefix.

Example:

```
Router# show ip cef 192.168.5.0 detail
192.168.5.0/24, version 21, cached adjacency to POS7/2
  0 packets, 0 bytes, traffic_index 4
    next hop 10.14.1.1, POS7/2 via 10.14.1.0/30
    valid cached adjacency
```

Step 2

`show ip bgp [network] [network-mask] [longer-prefixes]`

Enter the `show ip bgp` command for the same prefix used in Step 1--192.168.5.0--to learn which community is assigned to this prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that the community of 100:197 is assigned to this prefix.

Example:

```
Router# show ip bgp 192.168.5.0
BGP routing table entry for 192.168.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    10.14.1.1 from 10.14.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
    Community: 100:197
```

Step 3

`show cef interface [type number] policy-statistics [input] [output]`

Enter the `show cef interface policy-statistics` command to display the per-interface traffic statistics.

In this example, the output shows the number of packets and bytes that have been assigned to each accounting bucket:

Example:

```
Router# show cef interface policy-statistics input
```
FastEthernet1/0/0 is up (if_number 6)
Corresponding hwidb fast_if_number 6
Corresponding hwidb firstsw->if_number 6
BGP based Policy accounting on input is enabled

<table>
<thead>
<tr>
<th>Index</th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9999</td>
<td>999900</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>1234</td>
<td>123400</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>1000</td>
<td>100000</td>
</tr>
<tr>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>52</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>5123</td>
<td>1198782</td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>59</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>62</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Step 4**  
**show cef interface [type number] [statistics] [detail]**

Enter the `show cef interface EXEC` command to display the state of BGP policy accounting on a specified interface.
In this example, the output shows that BGP policy accounting has been configured to be based on input traffic at Fast Ethernet interface 1/0/0:

Example:

Router# show cef interface Fast Ethernet 1/0/0
FastEthernet1/0/0 is up (if_number 6)
  Corresponding hwidb fast_if_number 6
  Corresponding hwidb firstsw->if_number 6
  Internet address is 10.1.1.1/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  BGP based policy accounting on input is enabled
  BGP based policy accounting on output is disabled
  Hardware idb is FastEthernet1/0/0 (6)
  Software idb is FastEthernet1/0/0 (6)
  Fast switching type 1, interface type 18
  IP Distributed CEF switching enabled
  IP Feature Fast switching turbo vector
  IP Feature CEF switching turbo vector
  Input fast flags 0x100, Output fast flags 0x0, Flags 0x0
  ifindex 7(7)
  Slot 1 Slot unit 0 VC -1
  Transmit limit accumulator 0xE8001A82 (0xE8001A82)
  IP MTU 1500

---

Configuration Examples for BGP PA Output Interface Accounting

- Specifying the Match Criteria for BGP Policy Accounting Example. page 421
- Classifying the IP Traffic and Enabling BGP Policy Accounting Example. page 422

Specifying the Match Criteria for BGP Policy Accounting Example

In the following example, BGP communities are specified in community lists, and a route map named set_bucket is configured to match each of the community lists to a specific accounting bucket using the set traffic-index command:

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
!
route-map set_bucket permit 10
  match community-list 30
  set traffic-index 2
!
route-map set_bucket permit 20
  match community-list 40
  set traffic-index 3
!```
route-map set_bucket permit 30
  match community-list 50
  set traffic-index 4
route-map set_bucket permit 40
  match community-list 60
  set traffic-index 5

Classifying the IP Traffic and Enabling BGP Policy Accounting Example

In the following example, BGP policy accounting is enabled on POS interface 2/0/0. The policy accounting criteria is based on the source address of the input traffic, and the `table-map` command is used to modify the bucket number when the IP routing table is updated with routes learned from BGP.

```
router bgp 65000
  table-map set_bucket
  network 10.15.1.0 mask 255.255.255.0
  neighbor 10.14.1.1 remote-as 65100
  
  ip classless
  ip bgp-community new-format

  interface POS2/0/0
  ip address 10.15.1.2 255.255.255.0
  bgp-policy accounting input source
  no keepalive
  crc 32
  clock source internal
```

Additional References

The following sections provide references related to the BGP policy accounting output interface accounting feature.

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS IP Switching Command Reference</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
</tbody>
</table>

**Standards**

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-BGP-POLICY-ACCOUNTING-MIB</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for BGP Policy Accounting Output Interface Accounting

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 25  Feature Information for BGP Policy Accounting Output Interface Accounting

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Policy Accounting</td>
<td>Cisco IOS XE Release 2.1</td>
<td>BGP policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>BGP Policy Accounting Output Interface Accounting</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified for this feature: bgp-policy, set traffic-index, show cef interface, show cef interface policy-statistics</td>
</tr>
<tr>
<td>SNMP Support for BGP Policy Accounting</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The CISCO-BGP-POLICY-ACCOUNTING-MIB was introduced. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Glossary

**AS** --autonomous system. An IP term to describe a routing domain that has its own independent routing policy and is administered by a single authority.

**BGP** --Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems.

**CEF** --Cisco Express Forwarding.

**dCEF** --distributed Cisco Express Forwarding.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party...
BGP Cost Community

The BGP Cost Community feature introduces the cost extended community attribute. The cost community is a non-transitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not to external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the best-path selection process by assigning cost values to specific routes.

In Cisco IOS XE Release 2.1 and later releases, support was introduced for mixed EIGRP MPLS VPN network topologies that contain VPN and backdoor links.

- Finding Feature Information, page 427
- Prerequisites for the BGP Cost Community Feature, page 427
- Restrictions for the BGP Cost Community Feature, page 427
- Information About the BGP Cost Community Feature, page 428
- How to Configure the BGP Cost Community Feature, page 431
- Configuration Examples for the BGP Cost Community Feature, page 433
- Where to Go Next, page 435
- Additional References, page 435
- Related Documents, page 435
- Feature Information for BGP Cost Community, page 437

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for the BGP Cost Community Feature

This document assumes that BGP is configured in your network and that peering has been established.

Restrictions for the BGP Cost Community Feature
The BGP Cost Community feature can be configured only within an autonomous system or confederation. The cost community is a non-transitive extended community that is passed to iBGP and confederation peers only and is not passed to eBGP peers.

The BGP Cost Community feature must be supported on all routers in the autonomous system or confederation before cost community filtering is configured. The cost community should be applied consistently throughout the local autonomous system or confederation to avoid potential routing loops.

Multiple cost community set clauses may be configured with the `set extcommunity cost` command in a single route map block or sequence. However, each set clause must be configured with a different ID value (0-255) for each point of insertion (POI). The ID value determines preference when all other attributes are equal. The lowest ID value is preferred.

**Information About the BGP Cost Community Feature**

- **BGP Cost Community Overview**, page 428
- **How the BGP Cost Community Influences the Best Path Selection Process**, page 428
- **Cost Community Support for Aggregate Routes and Multipaths**, page 429
- **Influencing Route Preference in a Multi-Exit IGP Network**, page 430
- **BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links**, page 430

**BGP Cost Community Overview**

The cost community is a non-transitive extended community attribute that is passed to iBGP and confederation peers but not to eBGP peers. The configuration of the BGP Cost Community feature allows you to customize the BGP best path selection process for a local autonomous system or confederation.

The cost community attribute is applied to internal routes by configuring the `set extcommunity cost` command in a route map. The cost community set clause is configured with a cost community ID number (0-255) and cost number (0-4294967295). The cost number value determines the preference for the path. The path with the lowest cost community number is preferred. Paths that are not specifically configured with the cost community attribute are assigned a default cost number value of 2147483647 (The midpoint between 0 and 4294967295) and evaluated by the best path selection process accordingly. In the case where two paths have been configured with the same cost number value, the path selection process will then prefer the path with the lowest cost community ID. The cost extended community attribute is propagated to iBGP peers when extended community exchange is enabled with the `neighbor send-community` command.

The following commands can be used to apply the route map that is configured with the cost community set clause:

- aggregate-address
- neighbor default-originate route-map {in | out}
- neighbor route-map
- network route-map
- redistribute route-map

**How the BGP Cost Community Influences the Best Path Selection Process**

The cost community attribute influences the BGP best path selection process at the point of insertion (POI). By default, the POI follows the IGP metric comparison. When BGP receives multiple paths to the same
destination, it uses the best path selection process to determine which path is the best path. BGP automatically makes the decision and installs the best path into the routing table. The POI allows you to assign a preference to a specific path when multiple equal cost paths are available. If the POI is not valid for local best path selection, the cost community attribute is silently ignored.

Multiple paths can be configured with the cost community attribute for the same POI. The path with the lowest cost community ID is considered first. In other words, all of the cost community paths for a specific POI are considered, starting with the one with the lowest cost community. Paths that do not contain the cost community (for the POI and community ID being evaluated) are assigned the default community cost value (2147483647). If the cost community values are equal, then cost community comparison proceeds to the next lowest community ID for this POI.

Note

Paths that are not configured with the cost community attribute are considered by the best path selection process to have the default cost-value (half of the maximum value [4294967295] or 2147483647).

Applying the cost community attribute at the POI allows you to assign a value to a path originated or learned by a peer in any part of the local autonomous system or confederation. The cost community can be used as a “tie breaker” during the best path selection process. Multiple instances of the cost community can be configured for separate equal cost paths within the same autonomous system or confederation. For example, a lower cost community value can be applied to a specific exit path in a network with multiple equal cost exits points, and the specific exit path will be preferred by the BGP best path selection process. See the scenario described in "Influencing Route Preference in a Multi-Exit IGP Network".

Cost Community Support for Aggregate Routes and Multipaths

Aggregate routes and multipaths are supported by the BGP Cost Community feature. The cost community attribute can be applied to either type of route. The cost community attribute is passed to the aggregate or multipath route from component routes that carry the cost community attribute. Only unique IDs are passed, and only the highest cost of any individual component route will be applied to the aggregate on a per-ID basis. If multiple component routes contain the same ID, the highest configured cost is applied to the route. For example, the following two component routes are configured with the cost community attribute via an inbound route map:

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

If these component routes are aggregated or configured as a multipath, the cost value 200 (POI=IGP, ID=1, Cost=200) will be advertised because it is the highest cost.

If one or more component routes does not carry the cost community attribute or if the component routes are configured with different IDs, then the default value (2147483647) will be advertised for the aggregate or multipath route. For example, the following three component routes are configured with the cost community attribute via an inbound route map. However, the component routes are configured with two different IDs.

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 172.16.0.1 (POI=IGP, ID=2, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

The single advertised path will include the aggregated cost communities as follows:

- \{POI=IGP, ID=1, Cost=2147483647\} \{POI=IGP, ID=2, Cost=2147483647\}
Influencing Route Preference in a Multi-Exit IGP Network

The figure below shows an Interior Gateway Protocol (IGP) network with two autonomous system boundary routers (ASBRs) on the edge. Each ASBR has an equal cost path to network 10.8/16.

Figure 43  Multi-Exit Point IGP Network

Both paths are considered to be equal by BGP. If multipath loadsharing is configured, both paths will be installed to the routing table and will be used to load balance traffic. If multipath load balancing is not configured, then BGP will select the path that was learned first as the best path and install this path to the routing table. This behavior may not be desirable under some conditions. For example, the path is learned from ISP1 PE2 first, but the link between ISP1 PE2 and ASBR1 is a low-speed link.

The configuration of the cost community attribute can be used to influence the BGP best path selection process by applying a lower cost community value to the path learned by ASBR2. For example, the following configuration is applied to ASBR2.

    route-map ISP2_PE1 permit 10
      set extcommunity cost 1 1
      match ip address 13
    !
    ip access-list 13 permit 10.8.0.0 0.0.255.255

The above route map applies a cost community number value of 1 to the 10.8.0.0 route. By default, the path learned from ASBR1 will be assigned a cost community value of 2147483647. Because the path learned from ASBR2 has lower cost community value, this path will be preferred.

BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links

Before EIGRP Site of Origin (SoO) BGP Cost Community support was introduced, BGP preferred locally sourced routes over routes learned from BGP peers. Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. (A back door link, or a route, is a connection that is configured outside of the VPN between a remote and main site. For example, a WAN leased line that connects a remote site to the corporate network).

The "pre-best-path" point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP. The "pre-best path" POI carries the EIGRP
route type and metric. This POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required.

How to Configure the BGP Cost Community Feature

- Configuring the BGP Cost Community, page 431
- Verifying the Configuration of the BGP Cost Community, page 433

Configuring the BGP Cost Community

To configure the cost community, perform the task in this section.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. neighbor ip-address remote-as autonomous-system-number
5. address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | ipv6 [multicast | unicast] | vpnv4 [unicast]
6. neighbor ip-address route-map map-name {in | out}
7. exit
8. route-map map-name {permit | deny} [sequence-number]
9. set extcommunity cost [igp] community-id cost-value
10. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables higher privilege levels, such as privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Establishes peering with the specified neighbor or peer-group.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 10.0.0.1 remote-as 101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family ipv4 [mdt</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address route-map map-name [in</td>
<td>out]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# neighbor 10.0.0.1 route-map MAP-NAME in</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits router configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> route-map map-name [permit</td>
<td>deny] [sequence-number]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# route-map MAP-NAME permit 10</td>
<td></td>
</tr>
</tbody>
</table>
**Command or Action** | **Purpose**
---|---
**Step 9** | set extcommunity cost [igp] community-id cost-value
Example: Router(config-route-map)# set extcommunity cost 1 100
| Creates a set clause to apply the cost community attribute.
| • Multiple cost community set clauses can be configured in each route map block or sequence. Each cost community set clause must have a different ID (0-255). The cost community set clause with the lowest cost-value is preferred by the best path selection process when all other attributes are equal.
| • Paths that are not configured with the cost community attribute will be assigned the default cost-value, which is half of the maximum value (4294967295) or 2147483647.

**Step 10** | end
Example: Router(config-route-map)# end
| Exits route map configuration mode and enters privileged EXEC mode.

---

Verifying the Configuration of the BGP Cost Community

BGP cost community configuration can be verified locally or for a specific neighbor. To verify the local configuration cost community, use the `show route-map` or `show running-config` command. To verify that a specific neighbor carries the cost community, use the `show ip bgp ip-address` command. The output from these commands displays the POI (IGP is the default POI), the configured ID, and configured cost. For large cost community values, the output from these commands will also show, with + and - values, the difference between the configured cost and the default cost. See "Verifying the Configuration of the BGP Cost Community" for specific example output.

- Troubleshooting Tips, page 433

Troubleshooting Tips

The `bgp bestpath cost-community ignore` command can be used to disable the evaluation of the cost community attribute to help isolate problems and troubleshoot issues that relate to BGP best path selection.

The `debug ip bgp updates` command can be used to print BGP update messages. The cost community extended community attribute will be displayed in the output of this command when received from a neighbor. A message will also be displayed if a non-transitive extended community if received from an external peer.

Configuration Examples for the BGP Cost Community Feature

- BGP Cost Community Configuration Example, page 434
- BGP Cost Community Verification Examples, page 434
BGP Cost Community Configuration Example

The following example configuration shows the configuration of the set extcommunity cost command. The following example applies the cost community ID of 1 and cost community value of 100 to routes that are permitted by the route map. This configuration will cause the best path selection process to prefer this route over other equal cost paths that were not permitted by this route map sequence.

```
Router(config)# router bgp 50000
Router(config-router)# neighbor 10.0.0.1 remote-as 50000
Router(config-router)# neighbor 10.0.0.1 update-source Loopback 0
Router(config-router)# address-family ipv4
Router(config-router-af)# neighbor 10.0.0.1 activate
Router(config-router-af)# neighbor 10.0.0.1 route-map COST1 in
Router(config-router-af)# neighbor 10.0.0.1 send-community both
Router(config-router-af)# exit
Router(config)# route-map COST1 permit 10
Router(config-route-map)# match ip-address 1
Router(config-route-map)# set extcommunity cost 1 100
```

BGP Cost Community Verification Examples

BGP cost community configuration can be verified locally or for a specific neighbor. To verify the local configuration cost community, use the `show route-map` or `show running-config` command. To verify that a specific neighbor carries the cost community, use the `show ip bgp ip-address` command.

The output of the `show route-map` command will display locally configured route-maps, match, set, continue clauses, and the status and configuration of the cost community attribute. The following sample output is similar to the output that will be displayed:

```
Router# show route-map
route-map COST1, permit, sequence 10
  Match clauses:
    as-path (as-path filter): 1
  Set clauses:
    extended community Cost:igp:1:100
    policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 20
  Match clauses:
    ip next-hop (access-lists): 2
  Set clauses:
    extended community Cost:igp:2:200
    policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 30
  Match clauses:
    interface FastEthernet0/0
    extcommunity (extcommunity-list filter):300
  Set clauses:
    extended community Cost:igp:3:300
    policy routing matches: 0 packets, 0 bytes
```

The following sample output shows locally configured routes with large cost community values:

```
Router# show route-map
route-map set-cost, permit, sequence 10
  Match clauses:
    Set clauses:
      RT:700:700 additive
      extended community Cost:igp:1:4294967295 (default+2147483648)
      Cost:igp:7:2147284648 (default-198999)
      policy routing matches: 0 packets, 0 bytes
```
The output of the `show running config` command will display match, set, and continue clauses that are configured within a route-map. The following sample output is filtered to show only the relevant part of the running configuration:

```
Router# show running-config | begin route-map
route-map COST1 permit 20
  match ip next-hop 2
  set extcommunity cost igp 2 200

route-map COST1 permit 30
  match interface FastEthernet0/0
  match extcommunity 300
  set extcommunity cost igp 3 300
```

The output of the `show ip bgp ip-address` command can be used to verify if a specific neighbor carries a path that is configured with the cost community attribute. The cost community attribute information is displayed in the "Extended Community" field. The POI, the cost community ID, and the cost community number value are displayed. The following sample output shows that neighbor 172.16.1.2 carries a cost community with an ID of 1 and a cost of 100:

```
Router# show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  2 2 2
  172.16.1.2 from 172.16.1.2 (172.16.1.2)
    Origin IGP, metric 0, localpref 100, valid, external, best
    Extended Community: Cost:igp:1:100
```

If the specified neighbor is configured with the default cost community number value or if the default value is assigned automatically for cost community evaluation, "default" with + and - values will be displayed after the cost community number value in the output.

**Where to Go Next**

For more information about the EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature, refer to the "EIGRP MPLS VPN PE-CE Site of Origin (SoO)" module.

**Additional References**

For additional information related to the BGP Cost Community feature, refer to the following references.

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Best Path Selection</td>
<td>BGP Best Path Selection Algorithm</td>
</tr>
<tr>
<td>EIGRP MPLS VPN PE-CE Site of Origin</td>
<td>&quot;EIGRP MPLS VPN PE-CE Site of Origin (SoO)&quot;</td>
</tr>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
</tbody>
</table>
### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported, and support for existing standards has not been modified.</td>
<td>--</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported, and support for existing MIBs has not been modified.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-retana-bgp-custom-decision-00.txt</td>
<td>BGP Custom Decision Process</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
<tr>
<td>documentation and tools for troubleshooting and resolving technical issues</td>
<td></td>
</tr>
<tr>
<td>with Cisco products and technologies.</td>
<td></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you</td>
<td></td>
</tr>
<tr>
<td>can subscribe to various services, such as the Product Alert Tool (accessed</td>
<td></td>
</tr>
<tr>
<td>from Field Notices), the Cisco Technical Services Newsletter, and Really</td>
<td></td>
</tr>
<tr>
<td>Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user</td>
<td></td>
</tr>
<tr>
<td>ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for BGP Cost Community

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.
The BGP Cost Community feature introduces the cost extended community attribute. The cost community is a non-transitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not to external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the best-path selection process by assigning cost values to specific routes.

This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

The following commands were introduced or modified: `bgp bestpath cost-community ignore`, `debug ip bgp updates`, and `set extcommunity cost`.

---

**Table 26 Feature Information for BGP Cost Community**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Cost Community</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Cost Community feature introduces the cost extended community attribute. The cost community is a non-transitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not to external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the best-path selection process by assigning cost values to specific routes. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: <code>bgp bestpath cost-community ignore</code>, <code>debug ip bgp updates</code>, and <code>set extcommunity cost</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links</td>
<td>Cisco IOS XE Release 2.1</td>
<td>Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. The &quot;pre-best-path&quot; point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP and the POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required. This feature is enabled automatically for EIGRP VPN sites when Cisco IOS XE Release 2.1 or later releases, is installed to a PE, CE, or back door router. This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Support for IP Prefix Import from Global Table into a VRF Table

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map.

- Finding Feature Information, page 441
- Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 441
- Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 442
- Information About BGP Support for IP Prefix Import from Global Table into a VRF Table, page 442
- How to Import IP Prefixes from Global Table into a VRF Table, page 443
- Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 449
- Additional References, page 451
- Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 452

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Border Gateway Protocol (BGP) peering sessions are established.
- CEF or dCEF (for distributed platforms) is enabled on all participating routers.
Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Only IPv4 unicast and multicast prefixes can be imported into a VRF with this feature.
- A maximum of five VRF instances per router can be created to import IPv4 prefixes from the global routing table.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a VPNv4 VRF.

Information About BGP Support for IP Prefix Import from Global Table into a VRF Table

- Importing IPv4 Prefixes into a VRF, page 442
- Black Hole Routing, page 442
- Classifying Global Traffic, page 442
- Unicast Reverse Path Forwarding, page 443

Importing IPv4 Prefixes into a VRF

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding instance (VRF) table using an import route map. This feature extends the functionality of VRF import-map configuration to allow IPv4 prefixes to be imported into a VRF based on a standard community. Both IPv4 unicast and multicast prefixes are supported. No Multiprotocol Label Switching (MPLS) or route target (import/export) configuration is required.

IP prefixes are defined as match criteria for the import map through standard Cisco filtering mechanisms. For example, an IP access-list, an IP prefix-list, or an IP as-path filter is created to define an IP prefix or IP prefix range, and then the prefix or prefixes are processed through a match clause in a route map. Prefixes that pass through the route map are imported into the specified VRF per the import map configuration.

Black Hole Routing

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be configured to support Black Hole Routing (BHR). BHR is a method that allows the administrator to block undesirable traffic, such as traffic from illegal sources or traffic generated by a Denial of Service (DoS) attack, by dynamically routing the traffic to a dead interface or to a host designed to collect information for investigation, mitigating the impact of the attack on the network. Prefixes are looked up, and packets that come from unauthorized sources are blackholed by the ASIC at line rate.

Classifying Global Traffic

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be used to classify global IP traffic based on physical location or class of service. Traffic is classified based on administration policy and then imported into different VRFs. On a college campus, for example, network traffic could be divided into an academic network and residence network traffic, a student network and faculty network, or...
a dedicated network for multicast traffic. After the traffic is divided along administration policy, routing decisions can be configured with the MPLS VPN--VRF Selection Using Policy Based Routing feature or the MPLS VPN--VRF Selection Based on Source IP Address feature.

**Unicast Reverse Path Forwarding**

Unicast Reverse Path Forwarding (Unicast RPF) can be optionally configured with the BGP Support for IP Prefix Import from Global Table into a VRF Table feature. Unicast RPF is used to verify that the source address is in the Forwarding Information Base (FIB). The `ip verify unicast vrf` command is configured in interface configuration mode and is enabled for each VRF. This command has `permit` and `deny` keywords that are used to determine if the traffic is forwarded or dropped after Unicast RPF verification.

**How to Import IP Prefixes from Global Table into a VRF Table**

- Defining IPv4 IP Prefixes to Import, page 443
- Creating the VRF and the Import Route Map, page 444
- Filtering on the Ingress Interface, page 447
- Verifying Global IP Prefix Import, page 448

**Defining IPv4 IP Prefixes to Import**

IPv4 unicast or multicast prefixes are defined as match criteria for the import route map using standard Cisco filtering mechanisms. This task uses an IP access-list and an IP prefix-list.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `access-list access-list-number {deny | permit} source [source-wildcard] [log]`
4. `ip prefix-list prefix-list-name [seq seq-value] {deny network I length | permit network I length} [ge ge-value] [le le-value]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>
### Creating the VRF and the Import Route Map

The IP prefixes that are defined for import are then processed through a match clause in a route map. IP prefixes that pass through the route map are imported into the VRF. A maximum of 5 VRFs per router can be configured to import IPv4 prefixes from the global routing table. 1000 prefixes per VRF are imported by default. You can manually configure from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you manually configure the prefix import limit. Configuring the router to import too many prefixes can interrupt normal router operation.

No MPLS or route target (import/export) configuration is required.

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

The following syslog message is introduced by the BGP Support for IP Prefix Import from Global Table into a VRF Table feature. It will be displayed when more prefixes are available for import than the user-defined limit:

```
00:00:33: %BGP-3-APIMPORT_EXCEED: IPv4 Multicast prefixes imported to multicast vrf exceed the limit 2
```

You can either increase the prefix limit or fine-tune the import route map filter to reduce the number of candidate routes.

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> access-list access-list-number {deny</td>
<td>permit} source [source-wildcard] [log]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# access-list 50 permit 10.1.1.0 0.0.0.255</td>
<td>• The example creates a standard access list numbered 50. This filter will permit traffic from any host with an IP address in the 10.1.1.0/24 subnet.</td>
</tr>
<tr>
<td><strong>Step 4</strong> ip prefix-list prefix-list-name [seq seq-value] {deny network / length</td>
<td>permit network / length} [ge ge-value] [le le-value]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip prefix-list COLORADO permit 10.24.240.0/22</td>
<td>• The example creates an IP prefix list named COLORADO. This filter will permit traffic from any host with an IP address in the 10.24.240.0/22 subnet.</td>
</tr>
</tbody>
</table>
**Note**

- Only IPv4 unicast and multicast prefixes can be imported into a VRF with this feature.
- A maximum of five VRF instances per router can be created to import IPv4 prefixes from the global routing table.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a VPNv4 VRF.

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip vrf vrf-name</td>
<td>Creates a VRF routing table and specifies the VRF name (or tag).</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# ip vrf GREEN</td>
<td>• The <code>ip vrf vrf-name</code> command creates a VRF routing table and a CEF table, and both are named using the <code>vrf-name</code> argument. Associated with these tables is the default route distinguisher value.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Creates routing and forwarding tables for the VRF instance.</td>
</tr>
<tr>
<td><strong>rd route-distinguisher</strong></td>
<td>- There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# rd 100:10</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Creates an import map to import IPv4 prefixes from the global routing table to a VRF table.</td>
</tr>
<tr>
<td>**import ipv4 [unicast</td>
<td>multicast] [prefix-limit] map route-map**</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# import ipv4 unicast 1000 map UNICAST</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Exits VRF configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# exit</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.</td>
</tr>
<tr>
<td>**route-map map-tag [permit</td>
<td>deny] [sequence-number]**</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# route-map UNICAST permit 10</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>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.</td>
</tr>
<tr>
<td>**match ip address {acl-number [acl-number</td>
<td>acl-name]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# match ip address 50</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Exits route-map configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-route-map)# end</td>
</tr>
</tbody>
</table>
Filtering on the Ingress Interface

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be configured globally or on a per-interface basis. We recommend that you apply it to ingress interfaces to maximize performance.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number [name-tag]
4. ip policy route-map map-tag
5. ip verify unicast vrf vrf-name {deny | permit}
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type number [name-tag]</td>
<td>Configures an interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface Ethernet0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip policy route-map map-tag</td>
<td>Identifies a route map to use for policy routing on an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>• The example attaches the route map named UNICAST to the interface.</td>
</tr>
<tr>
<td>Router(config-if)# ip policy route-map UNICAST</td>
<td></td>
</tr>
</tbody>
</table>
Verifying Global IP Prefix Import

Perform the steps in this task to display information about the VRFs that are configured with the BGP Support for IP Prefix Import from Global Table into a VRF Table feature and to verify that global IP prefixes are imported into the specified VRF table.

**SUMMARY STEPS**

1. enable
2. show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name}
3. show ip vrf [brief | detail | interfaces | id] [vrf-name]

**DETAILED STEPS**

**Step 1**

**enable**

Enables privileged EXEC mode. Enter your password if prompted.

**Example:**

```
Router# enable
```

**Step 2**

**show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name}**

Displays VPN address information from the BGP table. The output displays the import route map, the traffic type (unicast or multicast), the default or user-defined prefix import limit, the actual number of prefixes that are imported, and individual import prefix entries.

**Example:**

```
Router# show ip bgp vpnv4 all
BGP table version is 15, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
```
Route Distinguisher: 100:1 (default for vrf academic)
Import Map: ACADEMIC, Address-Family: IPv4 Unicast, Pfx Count/Limit: 6/1000
* > 10.50.1.0/24     172.17.2.2                             0 2 3 ?
  > 10.50.2.0/24     172.17.2.2                             0 2 3 ?
  > 10.50.3.0/24     172.17.2.2                             0 2 3 ?
  > 10.60.1.0/24     172.17.2.2                             0 2 3 ?
  > 10.60.2.0/24     172.17.2.2                             0 2 3 ?
  > 10.60.3.0/24     172.17.2.2                             0 2 3 ?
Route Distinguisher: 200:1 (default for vrf residence)
Import Map: RESIDENCE, Address-Family: IPv4 Unicast, Pfx Count/Limit: 3/1000
* > 10.30.1.0/24     172.17.2.2                  0          0 2 i
  > 10.30.2.0/24     172.17.2.2                  0          0 2 i
  > 10.30.3.0/24     172.17.2.2                  0          0 2 i
Route Distinguisher: 300:1 (default for vrf BLACKHOLE)
Import Map: BLACKHOLE, Address-Family: IPv4 Unicast, Pfx Count/Limit: 3/1000
* > 10.40.1.0/24     172.17.2.2                  0          0 2 i
  > 10.40.2.0/24     172.17.2.2                  0          0 2 i
  > 10.40.3.0/24     172.17.2.2                  0          0 2 i
Route Distinguisher: 400:1 (default for vrf multicast)
Import Map: MCAST, Address-Family: IPv4 Multicast, Pfx Count/Limit: 2/2
* > 10.70.1.0/24     172.17.2.2                  0          0 2 i
  > 10.70.2.0/24     172.17.2.2                  0          0 2 i

**Step 3**

```plaintext
show ip vrf [brief | detail | interfaces | id] [vrf-name]
```

Displays defined VRFs and their associated interfaces. The output displays the import route map, the traffic type (unicast or multicast), and the default or user-defined prefix import limit. The following example output shows that the import route map named UNICAST is importing IPv4 unicast prefixes and that the prefix import limit is 1000.

**Example:**

```plaintext
Router# show ip vrf detail
VRF academic; default RD 100:10; default VPNID <not set>
VRF Table ID = 1
  No interfaces
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:10
  Import VPN route-target communities
    RT:100:10
  Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
  No export route-map
```

---

**Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table**

- Configuring Global IP Prefix Import Example, page 449
- Verifying Global IP Prefix Import Example, page 450

**Configuring Global IP Prefix Import Example**

The following example imports unicast prefixes into the VRF named `green` using an IP prefix list and a route map:

This example starts in global configuration mode:

```plaintext
```
Verifying Global IP Prefix Import Example

The `show ip vrf` command or the `show ip bgp vpnv4` command can be used to verify that prefixes are imported from the global routing table to the VRF table.

The following example from the `show ip vrf` command shows the import route map named UNICAST is importing IPv4 unicast prefixes and the prefix import limit is 1000:

```
Router# show ip vrf detail
VRF green; default RD 200:1; default VPNID <not set>
   Interfaces:
      Se2/0
   VRF Table ID = 1
      Export VPN route-target communities
         RT:200:10
      Import VPN route-target communities
         RT:200:10
      Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
         No export route-map
         VRF label distribution protocol: not configured
         VRF label allocation mode: per-prefix
      Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
         No export route-map
         VRF label distribution protocol: not configured
         VRF label allocation mode: per-prefix

VRF red; default RD 200:2; default VPNID <not set>
   Interfaces:
      Se3/0
   VRF Table ID = 2
      Export VPN route-target communities
         RT:200:20
      Import VPN route-target communities
         RT:200:20
      No import route-map
      No export route-map
      VRF label distribution protocol: not configured
      VRF label allocation mode: per-prefix
```

The following example from the `show ip bgp vpnv4` command shows the import route map names, the prefix import limit and the actual number of imported prefixes, and the individual import entries:

```
Router# show ip bgp vpnv4 all
BGP table version is 18, local router ID is 10.131.127.252
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf green)
Import Map: UNICAST, Address-Family: IPv4 Unicast, Pfx Count/Limit: 1/1000
   10.131.64.0/19  10.131.95.252  0 100 0 i
   172.16.1.1/32  172.16.2.1  0 32768 i
   172.16.2.0/30  0.0.0.0  0 32768 i
   172.31.1.1/32  10.131.95.252  0 100 0 i
   172.31.2.0/30  10.131.95.252  0 100 0 i
Route Distinguisher: 200:2 (default for vrf red)
```
### Additional References

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>MPLS Layer 3 VPN configuration tasks</td>
<td>&quot;Configuring MPLS Layer 3 VPNs&quot;</td>
</tr>
<tr>
<td>VRF selection based on source IP address</td>
<td>&quot;MPLS VPN--VRF Selection Based on Source IP Address&quot;</td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.
### Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Support for IP Prefix Import from Global Table into a VRF Table</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: <code>debug ip bgp import, import ipv4, ip verify unicast vrf</code>.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP per Neighbor SoO Configuration

The BGP per Neighbor SoO Configuration feature simplifies the configuration of the site-of-origin (SoO) value. Per neighbor SoO configuration introduces two new commands that can be configured in submodes under router configuration mode to set the SoO value.

- Finding Feature Information, page 455
- Prerequisites for BGP per Neighbor SoO Configuration, page 455
- Restrictions for BGP per Neighbor SoO Configuration, page 455
- Information About Configuring BGP per Neighbor SoO, page 456
- How to Configure BGP per Neighbor SoO, page 458
- Configuration Examples for BGP per Neighbor SoO Configuration, page 469
- Where to Go Next, page 470
- Additional References, page 470

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP per Neighbor SoO Configuration

This feature assumes that a Border Gateway Protocol (BGP) network is configured and that Cisco Express Forwarding is enabled in your network.

Restrictions for BGP per Neighbor SoO Configuration

A BGP neighbor or peer policy template-based SoO configuration takes precedence over the SoO value configured in an inbound route map.
Information About Configuring BGP per Neighbor SoO

- Site of Origin BGP Community Attribute, page 456
- Route Distinguisher, page 456
- BGP per Neighbor Site of Origin Configuration, page 456
- Benefits of BGP per Neighbor Site of Origin, page 457
- BGP Peer Policy Templates, page 457

Site of Origin BGP Community Attribute

The site-of-origin (SoO) extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a router has learned a route. BGP can use the SoO value associated with a route to prevent routing loops.

Route Distinguisher

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to change it into a globally unique VPN-IPv4 prefix. An RD can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:
  45000:3
- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:
  192.168.10.15:1

BGP per Neighbor Site of Origin Configuration

There are three ways to configure an SoO value for a BGP neighbor:

- BGP peer policy template--A peer policy template is created, and an SoO value is configured as part of the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.
- BGP neighbor command--Under address family IPv4 VRF, a neighbor is identified, and an SoO value is configured for the neighbor.
- BGP peer group--Under address family IPv4 VRF, a BGP peer group is configured, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

The configuration of SoO values for BGP neighbors is performed on a provider edge (PE) router, which is the VPN entry point. When SoO is enabled, the PE router forwards prefixes to the customer premises equipment (CPE) only when the SoO tag of the prefix does not match the SoO tag configured for the CPE. For example, in the figure below, an SoO tag is set as 65000:1 for the customer site that includes routers CPE1 and CPE2 with an autonomous system number of 65000. When CPE1 sends prefixes to PE1, PE1 tags the prefixes with 65000:1, which is the SoO tag for CPE1 and CPE2. When PE1 sends the tagged prefixes to PE2, PE2 performs a match against the SoO tag from CPE2. Any prefixes with the tag value of...
65000:1 are not sent to CPE2 because the SoO tag matches the SoO tag of CPE2, and a routing loop is avoided.

Figure 44  Network Diagram for SoO Example

Benefits of BGP per Neighbor Site of Origin

In releases prior to Cisco IOS Release 12.4(11)T, 12.2(33)SRB, and 12.2(33)SB, the SoO extended community attribute is configured using an inbound route map that sets the SoO value during the update process. The introduction of two new commands configured in submodes under router configuration mode simplifies the SoO value configuration.

BGP Peer Policy Templates

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address families. Peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.

Peer policy templates support inheritance. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group.

The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, very specific and very complex BGP policies can be created.

For more details about BGP peer policy templates, see the "Configuring a Basic BGP Network" module.
How to Configure BGP per Neighbor SoO

To configure an SoO value for a BGP neighbor, you must perform the first task in the following list and one of the next three tasks. The last three tasks are mutually exclusive; you need perform only one of them.

- Enabling Cisco Express Forwarding and Configuring VRF Instances, page 458
- Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template, page 461
- Configuring a per Neighbor SoO Value Using a BGP neighbor Command, page 464
- Configuring a per Neighbor SoO Value Using a BGP Peer Group, page 466

Enabling Cisco Express Forwarding and Configuring VRF Instances

Perform this task on both of the PE routers in the figure above to configure Virtual Routing and Forwarding (VRF) instances to be used with the per-VRF assignment tasks. In this task, Cisco Express Forwarding is enabled, and a VRF instance named SOO_VRF is created. To make the VRF functional, a route distinguisher is created, and the VRF is associated with an interface. When the route distinguisher is created, the routing and forwarding tables are created for the VRF instance named SOO_VRF. After associating the VRF with an interface, the interface is configured with an IP address.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip cef
4. ip vrf vrf-name
5. rd route-distinguisher
6. route-target {export | both} route-target-ext-community
7. route-target {import | both} route-target-ext-community
8. exit
9. interface type number
10. ip vrf forwarding vrf-name [downstream vrf-name2]
11. ip address ip-address mask [secondary]
12. end
13. show ip vrf [brief | detail | interfaces | id] [vrf-name] [output-modifiers]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip cef</td>
<td>Enables Cisco Express Forwarding on the route processor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip cef</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip vrf vrf-name</td>
<td>Defines a VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip vrf SOO_VRF</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> rd route-distinguisher</td>
<td>Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Router(config-vrf)# rd 1:1 | - Use the `route-distinguisher` argument to specify the default RD for a VPN. There are two formats that you can use to specify an RD:  
  - A 16-bit autonomous system number, a colon, and a 32-bit number, for example: `65000:3`  
  - A 32-bit IP address, a colon, and a 16-bit number, for example: `192.168.1.2:51`  
- In this example, the RD uses an autonomous system number with the number 1 after the colon. |
| **Step 6** route-target {export | both} route-target-ext-community | Creates a route-target extended community for a VRF. |
| **Example:** | |
| Router(config-vrf)# route-target export 1:1 | - Use the `export` keyword to export routing information to the target VPN extended community.  
- Use the `both` keyword to both import routing information from, and export routing information to, the target VPN extended community.  
- Use the `route-target-ext-community` argument to specify the VPN extended community. |

**Note** Only the syntax applicable to this step is displayed. For a different use of this syntax, see Step 7.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> route-target {import</td>
<td>both} route-target-ext-community</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# route-target import 1:1</td>
</tr>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Exits VRF configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# exit</td>
</tr>
<tr>
<td><strong>Step 9</strong> interface type number</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# interface GigabitEthernet 1/0/0</td>
</tr>
<tr>
<td><strong>Step 10</strong> ip vrf forwarding vrf-name [downstream vrf-name2]</td>
<td>Associates a VRF with an interface or subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# ip vrf forwarding SOO_VRF</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Executing this command on an interface removes the IP address, so the IP address should be reconfigured.</td>
</tr>
<tr>
<td><strong>Step 11</strong> ip address ip-address mask [secondary]</td>
<td>Configures an IP address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# ip address 192.168.1.2 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 12</strong> end</td>
<td>Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# end</td>
</tr>
</tbody>
</table>
**Command or Action**

<table>
<thead>
<tr>
<th>Step 13</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ip vrf [brief</td>
<td>detail</td>
<td>interfaces</td>
</tr>
</tbody>
</table>

- Use this command to verify the configuration of this task.

**Example:**

Router# show ip vrf

**Examples**

The following output of the `show ip vrf` command displays the VRF named SOO_VRF configured in this task.

```
Router# show ip vrf
Name                     Default RD  Interfaces
SOO_VRF                  1:1             GE1/0/0
```

**Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template**

Perform this task on router PE1 in the figure above to configure an SoO value for a BGP neighbor at the router CPE1 in the figure above using a peer policy template. In this task, a peer policy template is created, and the SoO value is configured for the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.

If a BGP peer inherits from several peer policy templates that specify different SoO values, the SoO value in the last template applied takes precedence and is applied to the peer. However, direct configuration of the SoO value on the BGP neighbor overrides any inherited template configurations of the SoO value.

This task assumes that the task described in the Enabling Cisco Express Forwarding and Configuring VRF Instances, page 458 has been performed.

**Note**

A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

>
### SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. template peer-policy policy-template-name
5. soo extended-community-value
6. exit-peer-policy
7. address-family ipv4 [unicast | multicast | vrf vrf-name]
8. neighbor ip-address remote-as autonomous-system-number
9. neighbor ip-address activate
10. neighbor ip-address inherit peer-policy policy-template-name
11. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> template peer-policy policy-template-name</td>
<td>Creates a peer policy template and enters policy-template configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# template peer-policy SOO_POLICY</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong> soo extended-community-value</td>
<td>Sets the SoO value for a BGP peer policy template.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-ptmp)# soo 65000:1</td>
<td>• Use the extended-community-value argument to specify the VPN extended community value. The value takes one of the following formats:</td>
</tr>
<tr>
<td></td>
<td>◦ A 16-bit autonomous system number, a colon, and a 32-bit number, for example: 45000:3</td>
</tr>
<tr>
<td></td>
<td>◦ A 32-bit IP address, a colon, and a 16-bit number, for example: 192.168.10.2:51</td>
</tr>
<tr>
<td></td>
<td>• In this example, the SoO value is set at 65000:1.</td>
</tr>
<tr>
<td><strong>Step 6</strong> exit-peer-policy</td>
<td>Exits policy-template configuration mode and returns to router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-pmtp)# exit-peer-policy</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> address-family ipv4 [unicast</td>
<td>multICAST] vrf vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router)# address-family ipv4 vrf SOO_VRF</td>
<td>• Use the unicast keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the unicast keyword is not specified with the address-family ipv4 command.</td>
</tr>
<tr>
<td></td>
<td>• Use the multicast keyword to specify IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• Use the vrf keyword and vrf-name argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
</tr>
<tr>
<td><strong>Step 8</strong> neighbor ip-address remote-as autonomous-system-number</td>
<td>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-af)# neighbor 192.168.1.1 remote-as 65000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor ip-address activate</td>
<td>Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-router-af)# neighbor 192.168.1.1 activate</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring a per Neighbor SoO Value Using a BGP neighbor Command

Perform this task on router PE2 in the figure above to configure an SoO value for the BGP neighbor at router CPE2 in the figure above using a `neighbor` command. Under address family IPv4 VRF, a neighbor is identified, and an SoO value is configured for the neighbor.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer policy template configurations of the SoO value.

For a configuration example involving 4-byte autonomous system numbers, see the GUID-9FA423D3-8CD0-4AAF-AEE2-BBB43D53575E.

This task assumes that the task described in the "Verifying CEF and Configuring VRF Instances" section has been performed with appropriate changes to interfaces and IP addresses.

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `address-family ipv4 [unicast | multicast] vrf vrf-name`
5. `neighbor [ip-address| peer-group-name] remote-as autonomous-system-number`
6. `neighbor ip-address activate`
7. `neighbor [ip-address| peer-group-name] soo extended-community-value`
8. `end`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted.  

**Example:**  
Router> enable |
| **Step 2** configure terminal | Enters global configuration mode.  

**Example:**  
Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process.  

**Example:**  
Router(config)# router bgp 50000 |
| **Step 4** address-family ipv4 [unicast | multicast | vrf vrf-name] | Specifies the IPv4 address family and enters address family configuration mode.  
- Use the **unicast** keyword to specify the IPv4 unicast address family.  
  By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.  
- Use the **multicast** keyword to specify IPv4 multicast address prefixes.  
- Use the **vrf** keyword and **vrf-name** argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.  

**Example:**  
Router(config-router)# address-family ipv4 vrf SOO_VRF |
| **Step 5** neighbor {ip-address | peer-group-name} remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.  

**Example:**  
Router(config-router-af)# neighbor 192.168.2.1 remote-as 65000 |
Configuring a per Neighbor SoO Value Using a BGP Peer Group

Perform this task on router PE1 in the figure above to configure an SoO value for the BGP neighbor at router CPE1 in the figure above using a neighbor command with a BGP peer group. Under address family IPv4 VRF, a BGP peer group is created and an SoO value is configured using a BGP neighbor command, and a neighbor is then identified and added as a peer group member. A BGP peer group member inherits the configuration associated with a peer group, which in this example, includes the SoO value.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer group configurations of the SoO value.

This task assumes that the task described in "Enabling Cisco Express Forwarding and Configuring VRF Instances" has been performed.

**Note**

A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.
SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 [unicast | multicast | vrf vrf-name]
5. neighbor peer-group-name peer-group
6. neighbor {ip-address | peer-group-name} soo extended-community-value
7. neighbor ip-address remote-as autonomous-system-number
8. neighbor ip-address activate
9. neighbor ip-address peer-group peer-group-name
10. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 50000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [unicast</td>
<td>multicast</td>
</tr>
<tr>
<td></td>
<td>configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>unicast</strong> keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the <strong>unicast</strong> keyword is not specified with the <strong>address-family ipv4</strong> command.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>multicast</strong> keyword to specify IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>vrf</strong> keyword and <strong>vrf-name</strong> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th><code>neighbor peer-group-name peer-group</code></th>
<th>Creates a BGP peer group.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-router-af)# neighbor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SOO_group peer-group</td>
<td></td>
</tr>
</tbody>
</table>
| Step 6 | `neighbor {ip-address peer-group-name} soo extended-community-value` | Sets the site-of-origin (SoO) value for a BGP neighbor or peer group.  
• In this example, the BGP peer group, SOO_group, is configured with an SoO value of 65000:1. |
|        | **Example:**                          |                           |
|        | Router(config-router-af)# neighbor    |                           |
|        | SOO_group soo 65000:1                 |                           |
| Step 7 | `neighbor ip-address remote-as autonomous-system-number` | Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. |
|        | **Example:**                          |                           |
|        | Router(config-router-af)# neighbor    |                           |
|        | 192.168.1.1 remote-as 65000           |                           |
| Step 8 | `neighbor ip-address activate`       | Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router. |
|        | **Example:**                          |                           |
|        | Router(config-router-af)# neighbor    |                           |
|        | 192.168.1.1 activate                 |                           |
| Step 9 | `neighbor ip-address peer-group peer-group-name` | Assigns the IP address of a BGP neighbor to a peer group. |
|        | **Example:**                          |                           |
|        | Router(config-router-af)# neighbor    |                           |
|        | 192.168.1.1 peer-group SOO_group      |                           |
| Step 10| `end`                                 | Exits address family configuration mode and returns to privileged EXEC mode. |
|        | **Example:**                          |                           |
|        | Router(config-router-af)# end         |                           |
Configuration Examples for BGP per Neighbor SoO Configuration

- Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template Example, page 469
- Configuring a per Neighbor SoO Value Using a BGP neighbor Command Example, page 469
- Configuring a per Neighbor SoO Value Using a BGP Peer Group Example, page 470

Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template Example

The following example shows how to create a peer policy template and configure an SoO value as part of the peer policy. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a peer policy template is created and an SoO value is configured as part of the peer policy. Under address family IPv4 VRF, a neighbor is identified and configured to inherit the peer policy that contains the SoO value.

```
ip cef
ip vrf SOO_VRF
rd 1:1
route-target export 1:1
route-target import 1:1
exit
interface GigabitEthernet 1/0/0
ip vrf forwarding SOO_VRF
ip address 192.168.1.2 255.255.255.0
exit
router bgp 50000
  address-family ipv4 vrf SOO_VRF
  neighbor 192.168.1.1 remote-as 65000
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 inherit peer-policy SOO_POLICY
end
```

Configuring a per Neighbor SoO Value Using a BGP neighbor Command Example

The following example shows how to configure an SoO value for a BGP neighbor. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a neighbor is identified under address family IPv4 VRF and an SoO value is configured for the neighbor.

```
ip cef
ip vrf SOO_VRF
rd 1:1
route-target export 1:1
route-target import 1:1
exit
interface GigabitEthernet 1/0/0
ip vrf forwarding SOO_VRF
ip address 192.168.2.2 255.255.255.0
exit
router bgp 50000
  address-family ipv4 vrf SOO_VRF
  neighbor 192.168.2.1 remote-as 65000
```

Configuration Examples for BGP per Neighbor SoO Configuration
neighbor 192.168.2.1 activate
neighbor 192.168.2.1 soo 65000:1
end

Configuring a per Neighbor SoO Value Using a BGP Peer Group Example

The following example shows how to configure an SoO value for a BGP peer group. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a BGP peer group is configured under address family IPv4 VRF, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

```
ip cef
ip vrf SOO_VRF
  rd 1:1
  route-target export 1:1
  route-target import 1:1
exit
interface GigabitEthernet 1/0/0
  ip vrf forwarding SOO_VRF
  ip address 192.168.1.2 255.255.255.0
exit
router bgp 50000
  address-family ipv4 vrf SOO_VRF
  neighbor SOO_GROUP peer-group
  neighbor SOO_GROUP soo 65000:65
  neighbor 192.168.1.1 remote-as 65000
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 peer-group SOO_GROUP
end
```
### Related Topic

<table>
<thead>
<tr>
<th>Document Title</th>
<th>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cisco IOS IP Routing: BGP Command Reference</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Title</th>
<th>IP Switching commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cisco IOS IP Switching Command Reference</strong></td>
<td></td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
</table>
| No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature. | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:  
http://www.cisco.com/go/mibs |

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

---

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Per-VRF Assignment of BGP Router ID

The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing `bgp router-id` command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

- Finding Feature Information, page 473
- Prerequisites for Per-VRF Assignment of BGP Router ID, page 473
- Information About Per-VRF Assignment of BGP Router ID, page 473
- How to Configure Per-VRF Assignment of BGP Router ID, page 474
- Configuration Examples for Per-VRF Assignment of BGP Router ID, page 491
- Additional References, page 497
- Feature Information for Per-VRF Assignment of BGP Router ID, page 498

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for Per-VRF Assignment of BGP Router ID

Before you configure this feature, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled in the network, and basic BGP peering is assumed to be running in the network.

Information About Per-VRF Assignment of BGP Router ID

- BGP Router ID, page 474
- Per-VRF Router ID Assignment, page 474
- Route Distinguisher, page 456
BGP Router ID

The BGP router identifier (ID) is a 4-byte field that is set to the highest IP address on the router. Loopback interface addresses are considered before physical interface addresses because loopback interfaces are more stable than physical interfaces. The BGP router ID is used in the BGP algorithm for determining the best path to a destination where the preference is for the BGP router with the lowest router ID. It is possible to manually configure the BGP router ID using the `bgp router-id` command to influence the best path algorithm.

Per-VRF Router ID Assignment

In Cisco IOS XE Release 2.1 and later releases, support for configuring separate router IDs for each Virtual Private Network (VPN) routing/forwarding (VRF) instance was introduced. The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing `bgp router-id` command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

Route Distinguisher

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to change it into a globally unique VPN-IPv4 prefix. An RD can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:
  45000:3
- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:
  192.168.10.15:1

How to Configure Per-VRF Assignment of BGP Router ID

There are two main ways to configure a BGP router ID for each separate VRF. To configure a per-VRF BGP router ID manually, you must perform the first three tasks listed below. To automatically assign a BGP router ID to each VRF, perform the first task and the fourth task.

- Configuring VRF Instances, page 474
- Associating VRF Instances with Interfaces, page 476
- Manually Configuring a BGP Router ID per VRF, page 479
- Automatically Assigning a BGP Router ID per VRF, page 484

Configuring VRF Instances

Perform this task to configure VRF instances to be used with the per-VRF assignment tasks. In this task, a VRF instance named vrf_trans is created. To make the VRF functional, a route distinguisher is created.
When the route distinguisher is created, the routing and forwarding tables are created for the VRF instance named vrf_trans.

This task assumes that you have Cisco Express Forwarding or distributed Cisco Express Forwarding enabled.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip vrf vrf-name`
4. `rd route-distinguisher`
5. `route-target {import | both} route-target-ext-community`
6. `route-target {export | both} route-target-ext-community`
7. `exit`
8. Repeat Step 3 through Step 7 for each VRF to be defined.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 enable</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2 configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3 ip vrf vrf-name</strong></td>
<td>Defines a VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# ip vrf vrf_trans</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4 rd route-distinguisher</strong></td>
<td>Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-vrf)# rd 45000:2</code></td>
<td>Use the <code>route-distinguisher</code> argument to specify the default RD for a VPN. There are two formats you can use to specify an RD. For more details, see the &quot;Route Distinguisher&quot; section.</td>
</tr>
<tr>
<td></td>
<td>In this example, the RD uses an autonomous system number with the number 2 after the colon.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5</td>
<td>`route-target {import</td>
<td>both} route-target-ext-community`</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>import</code> keyword to import routing information from the target</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VPN extended community.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>both</code> keyword to both import routing information from and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>export routing information to the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>route-target-ext-community</code> argument to specify the VPN</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>extended community.</td>
</tr>
<tr>
<td></td>
<td><code>Router(config-vrf)# route-target import 55000:5</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>`route-target {export</td>
<td>both} route-target-ext-community`</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>export</code> keyword to export routing information to the target</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VPN extended community.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>both</code> keyword to both import routing information from and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>export routing information to the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the <code>route-target-ext-community</code> argument to specify the VPN</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>extended community.</td>
</tr>
<tr>
<td></td>
<td><code>Router(config-vrf)# route-target export 55000:1</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>exit</code></td>
<td>Exits VRF configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td><code>Router(config-vrf)# exit</code></td>
</tr>
<tr>
<td>Step 8</td>
<td>Repeat Step 3</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>through Step 7</td>
<td>for each VRF to be defined.</td>
</tr>
</tbody>
</table>

## Associating VRF Instances with Interfaces

Perform this task to associate VRF instances with interfaces to be used with the per-VRF assignment tasks. In this task, a VRF instance named vrf_trans is associated with a serial interface.

Make a note of the IP addresses for any interface to which you want to associate a VRF instance because the `ip vrf forwarding` command removes the IP address. Step 8 allows you to reconfigure the IP address.

- This task assumes that you have Cisco Express Forwarding or distributed Cisco Express Forwarding enabled.
- This task assumes that VRF instances have been configured in the Configuring VRF Instances, page 474.
SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number
4. ip address ip-address mask [secondary]
5. exit
6. interface type number
7. ip vrf forwarding vrf-name [downstream vrf-name2]
8. ip address ip-address mask [secondary]
9. Repeat Step 5 through Step 8 for each VRF to be associated with an interface.
10. end
11. show ip vrf [brief | detail | interfaces | id] [vrf-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3 interface type number</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• In this example, loopback interface 0 is configured.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface loopback0</td>
</tr>
<tr>
<td>Step 4 ip address ip-address mask [secondary]</td>
<td>Configures an IP address.</td>
</tr>
<tr>
<td></td>
<td>• In this example, the loopback interface is configured with an IP address of 172.16.1.1.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)# ip address 172.16.1.1 255.255.255.255</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong> interface type number</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# interface serial2/0/0</td>
</tr>
<tr>
<td><strong>Step 7</strong> ip vrf forwarding vrf-name [downstream vrf-name2]</td>
<td>Associates a VRF with an interface or subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# ip vrf forwarding vrf_trans</td>
</tr>
<tr>
<td><strong>Step 8</strong> ip address ip-address mask [secondary]</td>
<td>Configures an IP address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# ip address 192.168.4.1 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 9</strong> Repeat Step 5 through Step 8 for each VRF to be associated with an interface.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 10</strong> end</td>
<td>Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)# end</td>
</tr>
<tr>
<td><strong>Step 11</strong> show ip vrf [brief</td>
<td>detail</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show ip vrf interfaces</td>
</tr>
</tbody>
</table>

**Examples**

The following output shows that two VRF instances named vrf_trans and vrf_users were configured on two serial interfaces.

Router# show ip vrf interfaces
Manually Configuring a BGP Router ID per VRF

Perform this task to manually configure a BGP router ID for each VRF. In this task, several address family configurations are shown and the router ID is configured in the IPv4 address family mode for one VRF instance. Step 22 shows you how to repeat certain steps to permit the configuration of more than one VRF on the same router.

This task assumes that you have previously created the VRF instances and associated them with interfaces. For more details, see the Configuring VRF Instances, page 474 and the Associating VRF Instances with Interfaces, page 476.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. no bgp default ipv4-unicast
5. bgp log-neighbor-changes
6. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
7. neighbor {ip-address|peer-group-name} update-source interface-type interface-number
8. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]}
9. neighbor {ip-address|peer-group-name} activate
10. neighbor {ip-address|peer-group-name} send-community {both|standard|extended}
11. exit-address-family
12. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]}
13. redistribute connected
14. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
15. neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]
16. neighbor {ip-address|peer-group-name} ebgp-multihop[ttl]
17. neighbor {ip-address|peer-group-name} activate
18. neighbor ip-address allowas-in [number]
19. no auto-summary
20. no synchronization
21. bgp router-id {ip-address|auto-assign}
22. Repeat Step 11 to Step 21 to configure another VRF instance.
23. end
24. show ip bgp vpnv4 {all|rd route-distinguisher|vrf vrf-name}
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp <em>autonomous-system-number</em></td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>Disables the IPv4 unicast address family for the BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# no bgp default ipv4-unicast</td>
<td>Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command. Existing neighbor configurations are not affected.</td>
</tr>
<tr>
<td><strong>Step 5</strong> bgp log-neighbor-changes</td>
<td>Enables logging of BGP neighbor resets.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# bgp log-neighbor-changes</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor {ip-address</td>
<td>peer-group-name} remote-as <em>autonomous-system-number</em></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-router)# neighbor 192.168.1.1 remote-as 45000</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command. Existing neighbor configurations are not affected.
### Per-VRF Assignment of BGP Router ID

#### How to Configure Per-VRF Assignment of BGP Router ID

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 7** neighbor \{ip-address| peer-group-name\} update-source interface-type interface-number | Allows BGP sessions to use any operational interface for TCP connections.  
- In this example, BGP TCP connections for the specified neighbor are sourced with the IP address of the loopback interface rather than the best local address. |
| **Example:**  
Router(config-router)# neighbor 192.168.1.1 update-source loopback0 | |
| **Step 8** address-family \{ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]\} | Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.  
- The example creates a VPNv4 address family session. |
| **Example:**  
Router(config-router)# address-family vpnv4 | |
| **Step 9** neighbor \{ip-address| peer-group-name\} activate | Activates the neighbor under the VPNv4 address family.  
- In this example, the neighbor 172.16.1.1 is activated. |
| **Example:**  
Router(config-router-af)# neighbor 172.16.1.1 activate | |
| **Step 10** neighbor \{ip-address| peer-group-name\} send-community \{both| standard| extended\} | Specifies that a communities attribute should be sent to a BGP neighbor.  
- In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1. |
| **Example:**  
Router(config-router-af)# neighbor 172.16.1.1 send-community extended | |
| **Step 11** exit-address-family | Exits address family configuration mode and returns to router configuration mode. |
| **Example:**  
Router(config-router-af)# exit-address-family | |
### Command or Action

**Step 12**  
*address-family [ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]]*

**Example:**

```
Router(config-router)# address-family ipv4 vrf vrf_trans
```

**Purpose**

Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.

- The example specifies that the VRF instance named vrf_trans is to be associated with subsequent IPv4 address family configuration commands.

**Step 13**  
*redistribute connected*

**Example:**

```
Router(config-router-af)# redistribute connected
```

**Purpose**

Redistributes from one routing domain into another routing domain.

- In this example, the `connected` keyword is used to represent routes that are established automatically when IP is enabled on an interface.
- Only the syntax applicable to this step is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

**Step 14**  
*neighbor {ip-address|peer-group-name} remote-as autonomous-system-number*

**Example:**

```
Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000
```

**Purpose**

Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

- If the `autonomous-system-number` argument matches the autonomous system number specified in the `router bgp` command, the neighbor is an internal neighbor.
- If the `autonomous-system-number` argument does not match the autonomous system number specified in the `router bgp` command, the neighbor is an external neighbor.
- In this example, the neighbor at 192.168.1.1 is an external neighbor.

**Step 15**  
*neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]*

**Example:**

```
Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend
```

**Purpose**

Customizes the AS_PATH attribute for routes received from an eBGP neighbor.

- The autonomous system number from the local BGP routing process is prepended to all external routes by default.
- Use the `no-prepend` keyword to not prepend the local autonomous system number to any routes received from the eBGP neighbor.
- In this example, routes from the neighbor at 192.168.1.1 will not contain the local autonomous system number.

**Step 16**  
*neighbor {ip-address|peer-group-name} ebgp-multihop[ttl]*

**Example:**

```
Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2
```

**Purpose**

Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

- In this example, BGP is configured to allow connections to or from neighbor 192.168.1.1, which resides on a network that is not directly connected.
### Command or Action

| Step 17 | `neighbor {ip-address|peer-group-name} activate` |
|---------|--------------------------------------------------|
| Purpose | Activates the neighbor under the IPV4 address family. |
|         | • In this example, the neighbor 192.168.1.1 is activated. |

**Example:**

```
Router(config-router-af)# neighbor 192.168.1.1 activate
```

<table>
<thead>
<tr>
<th>Step 18</th>
<th><code>neighbor ip-address allowas-in [number]</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers.</td>
</tr>
<tr>
<td></td>
<td>• In the example, the PE router with autonomous system number 45000 is configured to allow prefixes from the VRF vrf-trans. The neighboring PE router with the IP address 192.168.1.1 is set to be readvertised once to other PE routers with the same autonomous system number.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1
```

<table>
<thead>
<tr>
<th>Step 19</th>
<th><code>no auto-summary</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Disables automatic summarization and sends subprefix routing information across classful network boundaries.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# no auto-summary
```

<table>
<thead>
<tr>
<th>Step 20</th>
<th><code>no synchronization</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Enables the Cisco IOS XE software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# no synchronization
```

| Step 21 | `bgp router-id {ip-address|auto-assign}` |
|---------|-----------------------------------------|
| Purpose | Configures a fixed router ID for the local BGP routing process. |
|         | • In this example, the specified BGP router ID is assigned for the VRF instance associated with this IPv4 address family configuration. |

**Example:**

```
Router(config-router-af)# bgp router-id 10.99.1.1
```

<table>
<thead>
<tr>
<th>Step 22</th>
<th>Repeat Step 11 to Step 21 to configure another VRF instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 23</th>
<th><code>end</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-router-af)# end
```
### Command or Action

**Step 24**

```
show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name}
```

### Purpose

(Optional) Displays VPN address information from the BGP table.

- In this example, the complete VPNv4 database is displayed.

### Note

Only the syntax applicable to this task is used in this example. For more details, see the [Cisco IOS Multiprotocol Label Switching Command Reference](#).

### Examples

The following sample output assumes that two VRF instances named vrf_trans and vrf_user were configured each with a separate router ID. The router ID is shown next to the VRF name.

```text
Router# show ip bgp vpnv4 all
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network NEXT HOP   METRIC  LOCPRF WEIGHT PATH
Route Distinguisher: 1:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
  * 192.168.4.0    0.0.0.0          0  32768 ?
Route Distinguisher: 42:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
  *> 192.168.5.0   0.0.0.0           0  32768 ?
```

### Automatically Assigning a BGP Router ID per VRF

Perform this task to automatically assign a BGP router ID for each VRF. In this task, a loopback interface is associated with a VRF and the `bgp router-id` command is configured at the router configuration level to automatically assign a BGP router ID to all VRF instances. Step 9 shows you how to repeat certain steps to configure each VRF that is to be associated with an interface. Step 30 shows you how to configure more than one VRF on the same router.

This task assumes that you have previously created the VRF instances. For more details, see the [Configuring VRF Instances](#), page 474.
SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number
4. ip address ip-address mask [secondary]
5. exit
6. interface type number
7. ip vrf forwarding vrf-name [downstream vrf-name2]
8. ip address ip-address mask [secondary]
9. Repeat Step 5 through Step 8 for each VRF to be associated with an interface.
10. exit
11. router bgp autonomous-system-number
12. bgp router-id {ip-address} vrf auto-assign
13. no bgp default ipv4-unicast
14. bgp log-neighbor-changes
15. neighbor {ip-address} peer-group-name] remote-as autonomous-system-number
16. neighbor {ip-address} peer-group-name] update-source interface-type interface-number
17. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpn4 [unicast]}
18. neighbor {ip-address} peer-group-name] activate
19. neighbor {ip-address} peer-group-name] send-community {both| standard| extended}
20. exit-address-family
21. address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpn4 [unicast]}
22. redistribute connected
23. neighbor {ip-address} peer-group-name] remote-as autonomous-system-number
24. neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]
25. neighbor {ip-address} peer-group-name] bgp-multipath[ttl]
26. neighbor {ip-address} peer-group-name] activate
27. neighbor ip-address allowas-in [number]
28. no auto-summary
29. no synchronization
30. Repeat Step 20 to Step 29 to configure another VRF instance.
31. end
32. show ip bgp vpn4 {all} rd route-distinguisher| vrf vrf-name}
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Example:** | |
| Router> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | |
| Router# configure terminal | |
| **Step 3** interface type number | Configures an interface type and enters interface configuration mode.  
  • In this example, loopback interface 0 is configured. |
| **Example:** | |
| Router(config)# interface loopback0 | |
| **Step 4** ip address ip-address mask [secondary] | Configures an IP address.  
  • In this example, the loopback interface is configured with an IP address of 172.16.1.1. |
| **Example:** | |
| Router(config-if)# ip address 172.16.1.1 255.255.255.255 | |
| **Step 5** exit | Exits interface configuration mode and returns to global configuration mode. |
| **Example:** | |
| Router(config-if)# exit | |
| **Step 6** interface type number | Configures an interface type and enters interface configuration mode.  
  • In this example, loopback interface 1 is configured. |
| **Example:** | |
| Router(config)# interface loopback1 | |
| **Step 7** ip vrf forwarding vrf-name [downstream vrf-name] | Associates a VRF with an interface or subinterface.  
  • In this example, the VRF named vrf_trans is associated with loopback interface 1. |
| **Example:** | |
| Router(config-if)# ip vrf forwarding vrf_trans | **Note** Executing this command on an interface removes the IP address.  
  The IP address should be reconfigured. |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 8    | `ip address ip-address mask [secondary]` | Configures an IP address.  
  - In this example, loopback interface 1 is configured with an IP address of 10.99.1.1. |
| 9    | Repeat Step 5 through Step 8 for each VRF to be associated with an interface. | -- |
| 10   | `exit` | Exits interface configuration mode and returns to global configuration mode. |
| 11   | `router bgp autonomous-system-number` | Enters router configuration mode for the specified routing process. |
| 12   | `bgp router-id {ip-address| vrf auto-assign}` | Configures a fixed router ID for the local BGP routing process.  
  - In this example, a BGP router ID is automatically assigned for each VRF instance. |
| 13   | `no bgp default ipv4-unicast` | Disables the IPv4 unicast address family for the BGP routing process.  
  **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the `neighbor remote-as` router configuration command unless you configure the `no bgp default ipv4-unicast` router configuration command before configuring the `neighbor remote-as` command. Existing neighbor configurations are not affected. |
<p>| 14   | <code>bgp log-neighbor-changes</code> | Enables logging of BGP neighbor resets. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 15** neighbor {ip-address| peer-group-name} remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.  
- If the autonomous-system-number argument matches the autonomous system number specified in the `router bgp` command, the neighbor is an internal neighbor.  
- If the autonomous-system-number argument does not match the autonomous system number specified in the `router bgp` command, the neighbor is an external neighbor.  
- In this example, the neighbor is an internal neighbor. |
| **Step 16** neighbor {ip-address| peer-group-name} update-source interface-type interface-number | Allows BGP sessions to use any operational interface for TCP connections.  
- In this example, BGP TCP connections for the specified neighbor are sourced with the IP address of the loopback interface rather than the best local address. |
| **Step 17** address-family {ipv4 [mdt | multicast | unicast [vrf vrf-name] | vrf vrf-name] | vpnv4 [unicast]} | Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.  
- The example creates a VPNv4 address family session. |
| **Step 18** neighbor {ip-address| peer-group-name} activate | Activates the neighbor under the VPNv4 address family.  
- In this example, the neighbor 172.16.1.1 is activated. |
| **Step 19** neighbor {ip-address| peer-group-name} send-community {both | standard | extended} | Specifies that a communities attribute should be sent to a BGP neighbor.  
- In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1. |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>21</td>
<td>address-family {ipv4 [mdt</td>
</tr>
<tr>
<td>22</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>23</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>24</td>
<td>neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]</td>
</tr>
</tbody>
</table>

### Purpose

- Exits address family configuration mode and returns to router configuration mode.
- Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
- Redistributes from one routing domain into another routing domain.
- Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
- Customizes the AS_PATH attribute for routes received from an eBGP neighbor.

### Example

#### Step 20

```
Router(config-router-af)# exit-address-family
```

#### Step 21

```
Router(config-router)# address-family ipv4 vrf vrf_trans
```

- The example specifies that the VRF instance named vrf_trans is to be associated with subsequent IPv4 address family configuration mode commands.

#### Step 22

```
Router(config-router-af)# redistribute connected
```

- In this example, the `connected` keyword is used to represent routes that are established automatically when IP is enabled on an interface.
- Only the syntax applicable to this step is displayed. For more details, see the Cisco IOS IP Routing: BGP Command Reference.

#### Step 23

```
Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000
```

- If the `autonomous-system-number` argument matches the autonomous system number specified in the `router bgp` command, the neighbor is an internal neighbor.
- If the `autonomous-system-number` argument does not match the autonomous system number specified in the `router bgp` command, the neighbor is an external neighbor.
- In this example, the neighbor at 192.168.1.1 is an external neighbor.

#### Step 24

```
Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend
```

- The autonomous system number from the local BGP routing process is prepended to all external routes by default.
- Use the `no-prepend` keyword to not prepend the local autonomous system number to any routes received from the eBGP neighbor.
- In this example, routes from the neighbor at 192.168.1.1 will not contain the local autonomous system number.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 25   | `neighbor {ip-address|peer-group-name} ebgp-multihop[ttl]` | Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.  
• In this example, BGP is configured to allow connections to or from neighbor 192.168.1.1, which resides on a network that is not directly connected. |
|      | **Example:** |       |
|      | Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2 |       |
| 26   | `neighbor {ip-address|peer-group-name} activate` | Activates the neighbor under the IPV4 address family.  
• In this example, the neighbor 192.168.1.1 is activated. |
|      | **Example:** |       |
|      | Router(config-router-af)# neighbor 192.168.1.1 activate |       |
| 27   | `neighbor ip-address allowas-in [number]` | Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers.  
• In the example, the PE router with autonomous system number 45000 is configured to allow prefixes from the VRF vrf-trans. The neighboring PE router with the IP address 192.168.1.1 is set to be readvertised once to other PE routers with the same autonomous system number. |
|      | **Example:** |       |
|      | Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1 |       |
| 28   | `no auto-summary` | Disables automatic summarization and sends subprefix routing information across classful network boundaries. |
|      | **Example:** |       |
|      | Router(config-router-af)# no auto-summary |       |
| 29   | `no synchronization` | Enables the Cisco IOS XE software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP). |
|      | **Example:** |       |
|      | Router(config-router-af)# no synchronization |       |
| 30   | Repeat Step 20 to Step 29 to configure another VRF instance. | -- |
| 31   | `end` | Exits address family configuration mode and returns to privileged EXEC mode. |
|      | **Example:** |       |
|      | Router(config-router-af)# end |       |
**Step 32** show ip bgp vpnv4 {all| rd route-distinguisher| vrf vrf-name}

**Purpose**
(Optionalal) Displays VPN address information from the BGP table.
- In this example, the complete VPNv4 database is displayed.

**Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS Multiprotocol Label Switching Command Reference.

**Examples**
The following sample output assumes that two VRF instances named vrf_trans and vrf_user were configured, each with a separate router ID. The router ID is shown next to the VRF name.

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 172.16.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 172.22.0.0       0.0.0.0                  0         32768 ?
> 172.23.0.0       172.23.1.1               0             0 3 1 ?
*>10.21.1.1/32     192.168.3.1              0    100      0 2 i
*> 10.52.1.0/24     172.23.1.1               0             0 3 1 3 i
*> 10.52.2.1/32     172.23.1.1               0             0 3 1 3 i
*> 10.52.3.1/32     172.23.1.1               0             0 3 1 3 i
*> 10.99.1.1/32     172.23.1.1               0             0 3 1 ?
*> 10.99.1.2/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 10:1
*>10.21.1.1/32     192.168.3.1              0    100      0 2 i
Route Distinguisher: 42:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 2 i
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1               0             0 2 1 2 i
*>10.52.1.0/24     192.168.3.1              0    100      0 ?
*>10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.1.2/32     172.22.1.1               0             0 2 1 ?
```

**Configuration Examples for Per-VRF Assignment of BGP Router ID**

- Manually Configuring a BGP Router ID per VRF Examples, page 491
- Automatically Assigning a BGP Router ID per VRF Examples, page 493

**Manually Configuring a BGP Router ID per VRF Examples**
The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. The BGP router ID for each VRF is configured manually under separate IPv4 address families. The `show ip bgp vpnv4` command can be used to verify that the router IDs have been configured for each VRF. The configuration starts in global configuration mode.

```
ip vrf vrf_trans
```
rd 45000:1
route-target export 50000:50
route-target import 40000:1
!
ip vrf vrf_user
rd 65500:1
route-target export 65500:1
route-target import 65500:1
!
interface Loopback0
ip address 10.1.1.1 255.255.255.255
!
router bgp 45000
no bgp default ipv4-unicast
bgp log-neighbor-changes
neighbor 192.168.3.1 remote-as 45000
neighbor 192.168.3.1 update-source Loopback0
!
address-family vpnv4
neighbor 192.168.3.1 activate
neighbor 192.168.3.1 send-community extended
exit-address-family
!
address-family ipv4 vrf vrf_user
redistribute connected
neighbor 172.22.1.1 remote-as 40000
neighbor 172.22.1.1 local-as 50000 no-prepend
neighbor 172.22.1.1 ebgp-multihop 2
neighbor 172.22.1.1 activate
neighbor 172.22.1.1 allowas-in 1
no auto-summary
no synchronization
bgp router-id 10.99.1.1
exit-address-family
!
address-family ipv4 vrf vrf_trans
redistribute connected
neighbor 172.23.1.1 remote-as 50000
neighbor 172.23.1.1 local-as 40000 no-prepend
neighbor 172.23.1.1 ebgp-multihop 2
neighbor 172.23.1.1 activate
neighbor 172.23.1.1 allowas-in 1
no auto-summary
no synchronization
bgp router-id 10.99.1.2
exit-address-family

After the configuration, the output of the `show ip bgp vpnv4 all` command shows the router ID displayed next to the VRF name:

Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 172.23.0.0 172.23.1.1 0 32768
r> 172.22.0.0 0.0.0.0 0 32768
*>110.21.1.1/32 172.168.3.1 0 32768
*> 10.52.2.1/32 172.23.1.1 0 32768
*> 10.99.1.1/32 172.23.1.1 0 32768
*> 10.99.2.2/32 172.23.1.1 0 32768
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
*> 172.22.0.0 172.22.1.1 0 32768
r> 172.23.0.0 0.0.0.0 0 32768
*>110.21.1.1/32 172.168.3.1 0 32768
*> 10.52.2.1/32 172.22.1.1 0 32768
*>110.52.2.1/32 192.168.3.1 0 32768
The output of the `show ip bgp vpnv4 vrf` command for a specified VRF displays the router ID in the output header:

```
Router# show ip bgp vpnv4 vrf vrf_user
BGP table version is 43, local router ID is 10.99.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
  > 172.22.0.0       172.22.1.1               0             0 2 1 2
  > 10.21.1.1/32     172.22.1.1                             0 2 1 2 i
  > 10.99.1.1/32     0.0.0.0                  0         32768 ?
  > 10.99.2.2/32     172.22.1.1               0             0 2 1 2
```

The output of the `show ip bgp vpnv4 vrf summary` command for a specified VRF displays the router ID in the first line of the output:

```
Router# show ip bgp vpnv4 vrf vrf_user summary
BGP router identifier 10.99.1.1, local AS number 45000
BGP table version is 43, main routing table version 43
8 network entries using 1128 bytes of memory
8 path entries using 544 bytes of memory
16/10 BGP path/bestpath attribute entries using 1856 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
3 BGP extended community entries using 72 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
BGP using 3744 total bytes of memory
BGP activity 17/0 prefixes, 17/0 paths, scan interval 15 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
  172.22.1.1      4     2      20      21       43    0    0 00:12:33        3
```

When the path is sourced in the VRF, the correct router ID is displayed in the output of the `show ip bgp vpnv4 vrf` command for a specified VRF and network address:

```
Router# show ip bgp vpnv4 vrf vrf_user 172.23.0.0
BGP routing table entry for 65500:1:172.23.0.0/8, version 22
Paths: (1 available, best #1, table vrf_user)
  Advertised to update-groups:
    2
    3
  Local
  0.0.0.0 from 0.0.0.0 (10.99.1.1)
  Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
  Extended Community: RT:65500:1
```

Automatically Assigning a BGP Router ID per VRF Examples

The following three configuration examples show different methods of configuring BGP to automatically assign a separate router ID to each VRF instance:

- **Globally Automatically Assigned Router ID Using Loopback Interface IP Addresses Example**, page 494
- **Globally Automatically Assigned Router ID with No Default Router ID Example**, page 495
- **Per-VRF Automatically Assigned Router ID Example**, page 496
Globally Automatically Assigned Router ID Using Loopback Interface IP Addresses Example

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. Under router configuration mode, BGP is globally configured to automatically assign each VRF a BGP router ID. Loopback interfaces are associated with individual VRFs to source an IP address for the router ID. The `show ip bgp vpnv4` command can be used to verify that the router IDs have been configured for each VRF.

```
ip vrf vrf_trans
   rd 45000:1
   route-target export 50000:50
   route-target import 40000:1

ip vrf vrf_user
   rd 65500:1
   route-target export 65500:1
   route-target import 65500:1

interface Loopback0
   ip address 10.1.1.1 255.255.255.255

interface Loopback1
   ip vrf forwarding vrf_user
   ip address 10.99.1.1 255.255.255.255

interface Loopback2
   ip vrf forwarding vrf_trans
   ip address 10.99.2.2 255.255.255.255

router bgp 45000
   bgp router-id vrf auto-assign
   no bgp default ipv4-unicast
   bgp log-neighbor-changes
   neighbor 192.168.3.1 remote-as 45000
   neighbor 192.168.3.1 update-source Loopback0

   address-family vpnv4
      neighbor 192.168.3.1 activate
      neighbor 192.168.3.1 send-community extended
      exit-address-family

   address-family ipv4 vrf vrf_user
      redistribute connected
      neighbor 172.22.1.1 remote-as 40000
      neighbor 172.22.1.1 local-as 50000 no-prepend
      neighbor 172.22.1.1 ebgp-multihop 2
      neighbor 172.22.1.1 activate
      neighbor 172.22.1.1 allowas-in 1
      no auto-summary
      no synchronization
      exit-address-family

   address-family ipv4 vrf vrf_trans
      redistribute connected
      neighbor 172.23.1.1 remote-as 50000
      neighbor 172.23.1.1 local-as 2 no-prepend
      neighbor 172.23.1.1 ebgp-multihop 2
      neighbor 172.23.1.1 activate
      neighbor 172.23.1.1 allowas-in 1
      no auto-summary
      no synchronization
      exit-address-family
```

After the configuration, the output of the `show ip bgp vpnv4 all` command shows the router ID displayed next to the VRF name. Note that the router IDs used in this example are sourced from the IP addresses.
configured for loopback interface 1 and loopback interface 2. The router IDs are the same as in the
Manually Configuring a BGP Router ID per VRF Examples, page 491.

Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.2.2
*> 172.22.0.0 0.0.0.0 0 32768 ?
r> 172.23.0.0 172.23.1.1 0 0 3 1 ?
*10.21.1.1/32 192.168.3.1 0 100 0 2 i
> 10.52.1.0/24 172.23.1.1 0 3 1 ?
> 10.52.2.1/32 172.23.1.1 0 3 1 3 i
> 10.52.3.1/32 172.23.1.1 0 3 1 3 i
> 10.99.1.1/32 172.23.1.1 0 0 3 1 ?
> 10.99.1.2/32 0.0.0.0 0 32768 ?
Route Distinguisher: 50000:1
*10.21.1.1/32 192.168.3.1 0 100 0 2 i
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0 172.22.1.1 0 0 2 1 ?
*> 172.23.0.0 0.0.0.0 0 32768 ?
> 10.21.1.1/32 172.22.1.1 0 2 1 2 i
*10.52.1.0/24 192.168.3.1 0 100 0 ?
*10.52.2.1/32 192.168.3.1 0 100 0 3 i
*10.52.3.1/32 192.168.3.1 0 100 0 3 i
> 10.99.1.1/32 0.0.0.0 0 32768 ?
> 10.99.1.2/32 172.22.1.1 0 0 2 1 ?

Globally Automatically Assigned Router ID with No Default Router ID Example

The following example shows how to configure a router and associate a VRF that is automatically assigned
a BGP router ID when no default router ID is allocated.

```
ip vrf vpn1
  rd 45000:1
  route-target export 45000:1
  route-target import 45000:1
!
interface Loopback0
  ip vrf forwarding vpn1
  ip address 10.1.1.1 255.255.255.255
!
router bgp 45000
  bgp router-id vrf auto-assign
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  address-family ipv4 vrf vpn1
    neighbor 172.22.1.2 activate
    no auto-summary
    no synchronization
    exit-address-family
```

Assuming that a second router is configured to establish a session between the two routers, the output of the
show ip interface brief command shows only the VRF interfaces that are configured.

```
Router# show ip interface brief
Interface IP-Address OK? Method Status Protocol
Serial2/0/0 unassigned YES NVRAM administratively down down
Serial3/0/0 unassigned YES NVRAM administratively down down
Loopback0 10.1.1.1 YES NVRAM up up
```

The show ip vrf command can be used to verify that a router ID is assigned for the VRF:

```
Router# show ip vrf
```

Globally Automatically Assigned Router ID with No Default Router ID Example

The following example shows how to configure a router and associate a VRF that is automatically assigned
a BGP router ID when no default router ID is allocated.
Per-VRF Automatically Assigned Router ID Example

The following example shows how to configure two VRFs—vrf_trans and vrf_user—with sessions between each other on the same router. Under the IPv4 address family associated with an individual VRF, BGP is configured to automatically assign a BGP router ID. Loopback interfaces are associated with individual VRFs to source an IP address for the router ID. The output of the `show ip bgp vpnv4` command can be used to verify that the router IDs have been configured for each VRF.

```plaintext
ip vrf vrf_trans
  rd 45000:1
  route-target export 50000:50
  route-target import 40000:1
!
ip vrf vrf_user
  rd 65500:1
  route-target export 65500:1
  route-target import 65500:1
!
interface Loopback0
  ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
  ip vrf forwarding vrf_user
  ip address 10.99.1.1 255.255.255.255
!
interface Loopback2
  ip vrf forwarding vrf_trans
  ip address 10.99.2.2 255.255.255.255
!
routing bgp 45000
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 192.168.3.1 remote-as 45000
  neighbor 192.168.3.1 update-source Loopback0
!
address-family vpnv4
  neighbor 192.168.3.1 activate
  neighbor 192.168.3.1 send-community extended
  exit-address-family
!
address-family ipv4 vrf vrf_user
  redistribute connected
  neighbor 172.22.1.1 remote-as 40000
  neighbor 172.22.1.1 local-as 50000 no-prepend
  neighbor 172.22.1.1 ebgp-multihop 2
  neighbor 172.22.1.1 activate
  neighbor 172.22.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id auto-assign
  exit-address-family
!
address-family ipv4 vrf vrf_trans
  redistribute connected
  neighbor 172.23.1.1 remote-as 50000
  neighbor 172.23.1.1 local-as 40000 no-prepend
  neighbor 172.23.1.1 ebgp-multihop 2
  neighbor 172.23.1.1 activate
  neighbor 172.23.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id auto-assign
  exit-address-family
```

Per-VRF Automatically Assigned Router ID Example

VRF session is established:

<table>
<thead>
<tr>
<th>Name</th>
<th>Default RD</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpn1</td>
<td>45000:1</td>
<td>Loopback0</td>
</tr>
</tbody>
</table>
After the configuration, the output of the `show ip bgp vpnv4 all` command shows the router ID displayed next to the VRF name. Note that the router IDs used in this example are sourced from the IP addresses configured for loopback interface 1 and loopback interface 2.

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.2.2
  *> 172.22.0.0 0.0.0.0 0 32768 ?
  r> 172.23.0.0 172.23.1.1 0 0 3 1 ?
 *10.21.1.1/32 192.168.3.1 0 100 0 2 1
 *10.52.1.0/24 172.23.1.1 0 3 1 ?
 *10.52.2.1/32 172.23.1.1 0 3 1 3 i
 *10.52.3.1/32 172.23.1.1 0 3 1 3 i
 *10.99.1.1/32 172.23.1.1 0 0 3 1 ?
 *10.99.1.2/32 172.23.1.1 0 32768 ?
Route Distinguisher: 50000:1
  *10.21.1.1/32 192.168.3.1 0 100 0 2 i
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
  r> 172.22.0.0 172.22.1.1 0 0 2 1 ?
  *10.21.1.0/24 0.0.0.0 0 32768 ?
  *10.21.1.1/32 172.22.1.1 0 0 2 1 2 i
 *10.52.1.0/24 192.168.3.1 0 100 0 ?
 *10.52.2.1/32 192.168.3.1 0 100 0 3 i
 *10.52.3.1/32 192.168.3.1 0 100 0 3 i
 *10.99.1.1/32 172.22.1.1 0 0 32768 ?
 *10.99.1.2/32 172.22.1.1 0 0 2 1 ?
```

### Additional References

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>MPLS commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples</td>
<td><em>Cisco IOS Multiprotocol Label Switching Command Reference</em></td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>---</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for Per-VRF Assignment of BGP Router ID

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 28  Feature Information for Per-VRF Assignment of BGP Router ID

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-VRF Assignment of BGP Router ID</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing <code>bgp router-id</code> command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: <code>bgp router-id</code>, <code>show ip bgp vpnv4</code>.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Next Hop Unchanged

In an external BGP (eBGP) session, by default, the router changes the next hop attribute of a BGP route (to its own address) when the router sends out a route. The BGP Next Hop Unchanged feature allows BGP to send an update to an eBGP multihop peer with the next hop attribute unchanged.

- Finding Feature Information, page 501
- Information About Next Hop Unchanged, page 501
- How to Configure BGP Next Hop Unchanged, page 502
- Configuration Example for BGP Next Hop Unchanged, page 504
- Additional References, page 505
- Feature Information for BGP Next Hop Unchanged, page 505

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Next Hop Unchanged

- BGP Next Hop Unchanged, page 501

BGP Next Hop Unchanged

In an external BGP (eBGP) session, by default, the router changes the next hop attribute of a BGP route (to its own address) when the router sends out a route. If the BGP Next Hop Unchanged feature is configured, BGP will send routes to an eBGP multihop peer without modifying the next hop attribute. The next hop attribute is unchanged.
There is an exception to the default behavior of the router changing the next hop attribute of a BGP route when the router sends out a route. When the next hop is in the same subnet as the peering address of the eBGP peer, the next hop is not modified. This is referred to as third party next-hop.

The BGP Next Hop Unchanged feature provides flexibility when designing and migrating networks. It can be used only between eBGP peers configured as multihop. It can be used in a variety of scenarios between two autonomous systems. One scenario is when multiple autonomous systems are connected that share the same IGP, or at least the routers have another way to reach each other’s next hops (which is why the next hop can remain unchanged).

A common use of this feature is to configure Multiprotocol Label Switching (MPLS) inter-AS with multihop MP-eBGP for VPNv4 between RRs.

Another common use of this feature is a VPNv4 inter-AS Option C configuration, as defined in RFC4364, Section 10. In this configuration, VPNv4 routes are passed among autonomous systems between RR of different autonomous systems. The RRs are several hops apart, and have neighbor next-hop unchanged configured. PEs of different autonomous systems establish an LSP between them (via a common IGP or by advertising the next-hops--that lead to the PEs--via labeled routes among the ASBRs--routes from different autonomous systems separated by one hop). PEs are able to reach the next hops of the PEs in another AS via the LSPs, and can therefore install the VPNv4 routes in the VRF RIB.

Restriction

The BGP Next Hop Unchanged feature can be configured only between multihop eBGP peers. The following error message will be displayed if you try to configure this feature for a directly connected neighbor:

%BGP: Can propagate the nexthop only to multi-hop EBGP neighbor

How to Configure BGP Next Hop Unchanged

- Configuring the BGP Next Hop Unchanged for an eBGP Peer, page 503
Configuring the BGP Next Hop Unchanged for an eBGP Peer

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp as-number
4. address-family {ipv4 [unicast] | vpnv4 [unicast]}
5. neighbor ip-address remote-as as-number
6. neighbor ip-address activate
7. neighbor ip-address ebgp-multihop ttl
8. neighbor ip-address next-hop-unchanged
9. end
10. show ip bgp

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| Step 2          | configure terminal      | Enters global configuration mode.                                       |
|                 | Example:                |                                                                         |
|                 | Router# configure terminal |                                                                       |

| Step 3          | router bgp as-number   | Enters router configuration mode, and creates a BGP routing process.    |
|                 | Example:               |                                                                         |
|                 | Router(config)# router bgp 65535 |                                                                       |

<p>| Step 4          | address-family {ipv4 [unicast] | vpnv4 [unicast]} | Enters address family configuration mode to configure BGP peers to accept address family specific configurations. |
|                 | Example:                |                                                                         |
|                 | Router(config-router-af)# address-family vpnv4 |                                                                       |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> neighbor ip-address remote-as as-number</td>
<td>Adds an entry to the BGP neighbor table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.100 remote-as 65600</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor ip-address activate</td>
<td>Enables the exchange of information with the peer.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.100 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor ip-address ebgp-multihop ttl</td>
<td>Configures the local router to accept and initiate connections to external peers that reside on networks that are not directly connected.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> neighbor ip-address next-hop-unchanged</td>
<td>Configures the router to send BGP updates to the specified eBGP peer without modifying the next hop attribute.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# neighbor 10.0.0.100 next-hop-unchanged</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> end</td>
<td>Exits address family configuration mode, and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-router-af)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> show ip bgp</td>
<td>(Optional) Displays entries in the BGP routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show ip bgp</td>
<td>• The output will indicate if the neighbor next-hop-unchanged command has been configured for the selected address.</td>
</tr>
</tbody>
</table>

**Configuration Example for BGP Next Hop Unchanged**

- Example BGP Next Hop Unchanged for an eBGP Peer, page 505
Example BGP Next Hop Unchanged for an eBGP Peer

The following example configures a multihop eBGP peer at 10.0.0.100 in a remote AS. When the local router sends updates to that peer, it will send them without modifying the next hop attribute.

```
router bgp 65535
  address-family ipv4
  neighbor 10.0.0.100 remote-as 65600
  neighbor 10.0.0.100 activate
  neighbor 10.0.0.100 ebgp-multihop 255
  neighbor 10.0.0.100 next-hop-unchanged
end
```

Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>BGP configuration tasks</td>
<td>IP Routing: BGP Configuration Guide</td>
</tr>
<tr>
<td>MPLS configuration tasks</td>
<td>MPLS Configuration Guide</td>
</tr>
<tr>
<td>BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer</td>
<td>&quot;Configuring Internal BGP Features&quot;</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
</table>

Feature Information for BGP Next Hop Unchanged

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 29  Feature Information for BGP Next Hop Unchanged

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Next Hop Unchanged</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The BGP Next Hop Unchanged feature allows BGP to send an update to an eBGP multihop peer with the next hop attribute unchanged. The following command was added by this feature: <code>neighbor next-hop-unchanged.</code></td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Support for the L2VPN Address Family

BGP support for the Layer 2 Virtual Private Network (L2VPN) address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

- Finding Feature Information, page 507
- Prerequisites for BGP Support for the L2VPN Address Family, page 507
- Restrictions for BGP Support for the L2VPN Address Family, page 507
- Information About BGP Support for the L2VPN Address Family, page 508
- How to Configure BGP Support for the L2VPN Address Family, page 509
- Configuration Examples for BGP Support for the L2VPN Address Family, page 515
- Where to Go Next, page 518
- Additional References, page 518
- Feature Information for BGP Support for the L2VPN Address Family, page 519

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP Support for the L2VPN Address Family

The BGP Support for L2VPN Address Family feature assumes prior knowledge of Virtual Private Network (VPN), Virtual Private LAN Service (VPLS), and Multiprotocol Layer Switching (MPLS) technologies.

Restrictions for BGP Support for the L2VPN Address Family
For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

Information About BGP Support for the L2VPN Address Family

- L2VPN Address Family, page 508
- VPLS ID, page 509

L2VPN Address Family

In Cisco IOS XE Release 2.6 and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or Generic Routing Encapsulation (GRE). The L2VPN address family is configured under BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS Virtual Private LAN Service (VPLS) feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the "VPLS Autodiscovery: BGP Based" feature.

In L2VPN address family, the following BGP commands are supported:

- bgp next-hop
- bgp scan-time
- neighbor activate
- neighbor advertisement-interval
- neighbor allowas-in
- neighbor capability
- neighbor inherit
- neighbor maximum-prefix
- neighbor next-hop-self
- neighbor next-hop-unchanged
- neighbor peer-group
- neighbor remove-private-as
- neighbor route-map
- neighbor route-reflector-client
- neighbor send-community
- neighbor soft-reconfiguration
• neighbor soo
• neighbor weight

Note

For route reflectors using L2VPNs, the neighbor next-hop-self and neighbor next-hop-unchanged commands are not supported.

For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

VPLS ID

A VPLS ID is a BGP extended community value that identifies the VPLS domain. Manual configuration of this ID is optional because a default VPLS ID is generated using the BGP autonomous system number and the configured VPN ID. A VPLS ID can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter a VPLS ID in either of these formats:

• Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:
  45000:3
• Enter a 32-bit IP address, a colon, and a 16-bit number. For example:
  192.168.10.15:1

How to Configure BGP Support for the L2VPN Address Family

• Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family, page 509

Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family

Perform this task to implement VPLS autodiscovery of each provider edge (PE) router that is a member of a specific VPLS. In Cisco IOS XE Release 2.6, the BGP L2VPN address family was introduced with a separate L2VPN RIB that contains endpoint provisioning information. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time any Layer 2 (L2) virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

BGP-based VPLS autodiscovery eliminates the need to manually provision a VPLS neighbor. After a PE router configures itself to be a member of a particular VPLS, information needed to set up connections to remote routers in the same VPLS is distributed by a discovery process. When the discovery process is complete, each member of the VPLS will have the information needed to set up VPLS pseudowires to form the full mesh of pseudowires needed for the VPLS.
This task is configured at router N-PE3 in the figure below and must be repeated at routers N-PE1 and N-PE2 with the appropriate changes such as different IP addresses. For a full configuration of these routers, see the figure below.

Figure 45  Network Diagram for BGP Autodiscovery Using the L2VPN Address Family

MISSING ILLO. EMBEDDED, NOT REFERENCED.

In this task, the PE router N-PE3 in the figure above is configured with a Layer 2 router ID, a VPN ID, a VPLS ID, and is enabled to automatically discover other PE routers that are part of the same VPLS domain. A BGP session is created to activate BGP neighbors under the L2VPN address family. Finally, two optional show commands are entered to verify the steps in the task.

This task assumes that MPLS is configured with VPLS options. For more details, see the “VPLS Autodiscovery: BGP Based” feature.

SUMMARY STEPS

1. enable
2. configure terminal
3. l2 router-id ip-address
4. l2 vfi vfi-name autodiscovery
5. vpn id vpn-id
6. vpls-id vpls-id
7. exit
8. Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.
9. router bgp autonomous-system-number
10. no bgp default ipv4-unicast
11. bgp log-neighbor-changes
12. bgp update-delay seconds
13. neighbor {ip-address|peer-group-name} remote-as autonomous-system-number
14. neighbor {ip-address|peer-group-name} update-source interface-type interface-number
15. Repeat Step 13 and Step 14 to configure other BGP neighbors.
16. address-family l2vpn [vpls]
17. neighbor ip-address activate
18. neighbor {ip-address|peer-group-name} send-community {both | standard | extended}
19. Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.
20. end
21. show vfi
22. show ip bgp l2vpn vpls {all | rd vpn-rd}
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal |
| **Step 3** 12 router-id *ip-address* | Specifies a router ID (in IP address format) for the PE router to use with VPLS autodiscovery pseudowires.  
  - In this example, the L2 router ID is defined as 10.1.1.3. |
| **Example:** Router(config)# 12 router-id 10.1.1.3 |
| **Step 4** 12 vfi vfi-name autodiscovery | Creates an L2 VFI, enables the VPLS PE router to automatically discover other PE routers that are part of the same VPLS domain, and enters L2 VFI autodiscovery configuration mode.  
  - In this example, the L2 VFI named customerA is created. |
| **Example:** Router(config)# 12 vfi customerA autodiscovery |
| **Step 5** vpn id *vpn-id* | Specifies a VPN ID.  
  - Use the same VPN ID for the PE routers that belong to the same VPN. Make sure that the VPN ID is unique for each VPN in the service provider network.  
  - Use the *vpn-id* argument to specify a number in the range from 1 to 4294967295.  
  - In this example, a VPN ID of 100 is specified. |
| **Example:** Router(config-vfi)# vpn id 100 |
| **Step 6** vpls-id *vpls-id* | (Optional) Specifies a VPLS ID.  
  - The VPLS ID is an identifier that is used to identify the VPLS domain. This command is optional because a default VPLS ID is automatically generated using the BGP autonomous system number and the VPN ID configured for the VFI. Only one VPLS ID can be configured per VFI, and the same VPLS ID cannot be configured in multiple VFIs on the same router.  
  - In this example, a VPLS ID of 65000:100 is specified. |
| **Example:** Router(config-vfi)# vpls-id 65000:100 |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>exit</td>
<td>Exits L2 VFI autodiscovery configuration mode and returns to global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
Router(config-vfi)# exit
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
Router(config)# router bgp 65000
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td><code>no bgp default ipv4-unicast</code></td>
<td>Disables the IPv4 unicast address family for the BGP routing process.</td>
</tr>
</tbody>
</table>

**Note**

Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the `neighbor remote-as` router configuration command unless you configure the `no bgp default ipv4-unicast` router configuration command before configuring the `neighbor remote-as` command. Existing neighbor configurations are not affected.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td><code>bgp log-neighbor-changes</code></td>
<td>Enables logging of BGP neighbor resets.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
Router(config-router)# bgp log-neighbor-changes
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td><code>bgp update-delay seconds</code></td>
<td>Sets the maximum initial delay period before a BGP-speaking networking device sends its first updates.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
Router(config-router)# bgp update-delay 1
```

• Use the `seconds` argument to set the delay period.
### Command or Action

**Step 13** `neighbor {ip-address|peer-group-name} remote-as autonomous-system-number`  
**Example:**
```
Router(config-router)# neighbor 10.10.10.1 remote-as 65000
```
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• If the <code>autonomous-system-number</code> argument matches the autonomous system number specified in the <code>router bgp</code> command, the neighbor is an internal neighbor.</td>
</tr>
<tr>
<td></td>
<td>• If the <code>autonomous-system-number</code> argument does not match the autonomous system number specified in the <code>router bgp</code> command, the neighbor is an external neighbor.</td>
</tr>
<tr>
<td></td>
<td>• In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.</td>
</tr>
</tbody>
</table>

**Step 14** `neighbor {ip-address|peer-group-name} update-source interface-type interface-number`  
**Example:**
```
Router(config-router)# neighbor 10.10.10.1 update-source loopback 1
```
<table>
<thead>
<tr>
<th>Purpose</th>
<th>(Optional) Configures a router to select a specific source or interface to receive routing table updates.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• This example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface.</td>
</tr>
</tbody>
</table>

**Step 15** Repeat Step 13 and Step 14 to configure other BGP neighbors.  

**Step 16** `address-family l2vpn [vpls]`  
**Example:**
```
Router(config-router)# address-family l2vpn vpls
```
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Specifies the L2VPN address family and enters address family configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The optional <code>vpls</code> keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers.</td>
</tr>
<tr>
<td></td>
<td>• In this example, an L2VPN VPLS address family session is created.</td>
</tr>
</tbody>
</table>

**Step 17** `neighbor ip-address activate`  
**Example:**
```
Router(config-router-af)# neighbor 10.10.10.1 activate
```
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Enables the neighbor to exchange information for the L2VPN VPLS address family with the local router.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note</strong></td>
<td>If you have configured a BGP peer group as a neighbor, you do not use this step. BGP peer groups are activated when a BGP parameter is configured. For example, the <code>neighbor send-community</code> command in the next step will automatically activate a peer group.</td>
</tr>
</tbody>
</table>

**Step 18** `neighbor {ip-address|peer-group-name} send-community{both|standard|extended}`  
**Example:**
```
Router(config-router-af)# neighbor 10.10.10.1 send-community extended
```
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Specifies that a communities attribute should be sent to a BGP neighbor.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• In this example, an extended communities attribute is sent to the neighbor at 10.10.1.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 19</strong></td>
<td>Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.</td>
</tr>
<tr>
<td><strong>Step 20</strong> end</td>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-router-af)# end</td>
</tr>
<tr>
<td><strong>Step 21</strong> show vfi</td>
<td>(Optional) Displays information about the configured VFI instances.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show vfi</td>
</tr>
<tr>
<td><strong>Step 22</strong> show ip bgp l2vpn vpls {all</td>
<td>rd vpn-rd}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show ip bgp l2vpn vpls all</td>
</tr>
</tbody>
</table>

**Examples**

The following is sample output from the `show vfi` command that shows two VFIs, CustomerA and CustomerB, with their associated VPN and VPLS IDs:

```
Router# show vfi
Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No
VFI name: customerA, state: down, type: multipoint
  VPN ID: 100, VPLS-ID: 65000:100
  RD: 65000:100, RT: 65000:100
  Local attachment circuits:
    Neighbors connected via pseudowires:
      Peer Address     VC ID        Discovered Router ID    S
      10.10.10.1         100        10.10.10.99             Y
VFI name: customerB, state: down, type: multipoint
  VPN ID: 200, VPLS-ID: 65000:200
  RD: 65000:200, RT: 65000:200
  Local attachment circuits:
    Neighbors connected via pseudowires:
      Peer Address     VC ID        Discovered Router ID    S
      10.10.10.3         200                 10.10.10.98    Y
```

The following is sample output from the `show ip bgp l2vpn vpls all` command that shows two VFIs identified by their VPN route distinguisher:

```
Router# show ip bgp l2vpn vpls all
BGP table version is 5, local router ID is 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network   Next Hop   Metric   LocPrf   Weight   Path
          Peer Address     VC ID        Discovered Router ID    S
*> 65000:100:10.10.10.96/96 0.0.0.0 32768 ?
```
What to Do Next

To configure more VPLS features, see the main VPLS documentation in the "VPLS Autodiscovery: BGP Based" feature.

Configuration Examples for BGP Support for the L2VPN Address Family

Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example

In this configuration example, all the routers in autonomous system 65000 in the figure below are configured to provide BGP support for the L2VPN address family. VPLS autodiscovery is enabled and L2 VFI and VPN IDs are configured. BGP neighbors are configured and activated under L2VPN address family to ensure that the VPLS endpoint provisioning information is saved to a separate L2VPN RIB and then distributed to the other BGP peers in BGP update messages. When the endpoint information is received by the BGP peers, a pseudowire mesh is set up to support L2VPN-based services.

Figure 46  Network Diagram for VPLS Autodiscovery Using BGP and the L2VPN Address Family

- N-PE = Network-facing PE router
- U-PE = User-facing PE router
**Router N-PE1**

```conf
ip subnet-zero
ip cef
no ip dhcp use vrf connected
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 1000 2000
mpls label protocol ldp
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
vpn id 100

pseudowire-class mpls
  encapsulation mpls
  interface Loopback1
    ip address 10.1.1.1 255.255.255.255
  interface GigabitEthernet0/0/1
    description Backbone interface
    ip address 10.0.0.1 255.255.255.0
    mpls ip
  router ospf 1
    log-adjacency-changes
    network 10.10.1.0 0.0.0.255 area 0
    network 192.168.0.0 0.0.0.255 area 0
  router bgp 65000
    no bgp default ipv4-unicast
    bgp log-neighbor-changes
    bgp update-delay 1
    neighbor 10.10.10.2 remote-as 65000
    neighbor 10.10.10.2 update-source Loopback 1
    neighbor 10.10.10.3 remote-as 65000
    neighbor 10.10.10.3 update-source Loopback 1
  address-family l2vpn vpls
    neighbor 10.10.10.2 activate
    neighbor 10.10.10.2 send-community extended
    neighbor 10.10.10.3 activate
    neighbor 10.10.10.3 send-community extended
    exit-address-family
  ip classless
```

**Router N-PE2**

```conf
ip subnet-zero
ip cef
no ip dhcp use vrf connected
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
vpn id 100

pseudowire-class mpls
  encapsulation mpls
  interface Loopback1
    ip address 10.1.1.2 255.255.255.255
  interface GigabitEthernet0/0/1
    description Backbone interface
    ip address 10.0.0.2 255.255.255.0
    mpls ip
```

---

**IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2**

516
router ospf 1
  log-adjacency-changes
  network 10.10.1.0 0.0.0.255 area 0
  network 192.168.0.0 0.0.0.255 area 0

router bgp 65000
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  bgp update-delay 1
  neighbor 10.10.10.1 remote-as 65000
  neighbor 10.10.10.1 update-source Loopback1
  neighbor 10.10.10.3 remote-as 65000
  neighbor 10.10.10.3 update-source Loopback1

  address-family l2vpn vpls
  neighbor 10.10.10.1 activate
  neighbor 10.10.10.1 send-community extended
  neighbor 10.10.10.3 activate
  neighbor 10.10.10.3 send-community extended
  exit-address-family

  ip classless

Router N-PE3

ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
vpn id 100
!
pseudowire-class mpls
  encapsulation mpls
  !
  interface Loopback1
    ip address 10.1.1.3 255.255.255.255
    !
  interface GigabitEthernet0/0/1
    description Backbone Interface
    ip address 10.0.0.3 255.255.255.0
    mpls ip
    !
router ospf 1
  log-adjacency-changes
  network 10.10.1.0 0.0.0.255 area 0
  network 192.168.0.0 0.0.0.255 area 0

router bgp 65000
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  bgp update-delay 1
  neighbor 10.10.10.1 remote-as 65000
  neighbor 10.10.10.1 update-source Loopback1
  neighbor 10.10.10.2 remote-as 65000
  neighbor 10.10.10.2 update-source Loopback1

  address-family l2vpn vpls
  neighbor 10.10.10.1 activate
  neighbor 10.10.10.1 send-community extended
  neighbor 10.10.10.2 activate
  neighbor 10.10.10.2 send-community extended
  exit-address-family

  ip classless
Where to Go Next

For more details about configuring VPLS autodiscovery, see the "VPLS Autodiscovery: BGP Based" feature.

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>BGP overview</td>
<td>Cisco BGP Overview” module</td>
</tr>
<tr>
<td>Configuring basic BGP tasks</td>
<td>Configuring a Basic BGP Network” module</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL:</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including</td>
<td><a href="http://www.cisco.com/cisco/web/support/index">http://www.cisco.com/cisco/web/support/index</a></td>
</tr>
<tr>
<td>documentation and tools for troubleshooting and resolving technical issues</td>
<td></td>
</tr>
<tr>
<td>with Cisco products and technologies.</td>
<td></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can</td>
<td></td>
</tr>
<tr>
<td>subscribe to various services, such as the Product Alert Tool (accessed</td>
<td></td>
</tr>
<tr>
<td>from Field Notices), the Cisco Technical Services Newsletter, and Really</td>
<td></td>
</tr>
<tr>
<td>Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com</td>
<td></td>
</tr>
<tr>
<td>user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for BGP Support for the L2VPN Address Family

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>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Support for the L2VPN Address Family</td>
<td>Cisco IOS XE Release 2.6</td>
<td>BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services. The following commands were introduced or modified by this feature: <code>address-family l2vpn</code>, <code>clear ip bgp l2vpn</code>, and <code>show ip bgp l2vpn</code>.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP 4 MIB Support for Per-Peer Received Routes

This document describes BGP 4 MIB support for per-peer received routes. This feature introduces a table in the CISCO-BGP4-MIB that provides the capability to query (by using Simple Network Management Protocol [SNMP] commands) for routes that are learned from individual Border Gateway Protocol (BGP) peers.

- Finding Feature Information, page 521
- Restrictions on BGP 4 MIB Support for Per-Peer Received Routes, page 521
- Information About BGP 4 MIB Support for Per-Peer Received Routes, page 522
- Additional References, page 526
- Feature Information for BGP 4 MIB Support for Per-Peer Received Routes, page 527
- Glossary, page 527

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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.

Restrictions on BGP 4 MIB Support for Per-Peer Received Routes

BGP 4 MIB Support for per-Peer Received Routes supports only routes that are contained in IPv4 AFIs and unicast SAFIs in the local BGP RIB table. The BGP 4 MIB Support for per-Peer Received Routes enhancement is supported only by BGP Version 4.
Overview of BGP 4 MIB Support for Per-Peer Received Routes

The BGP 4 MIB support for per-peer received routes feature introduces a table in the CISCO-BGP4-MIB that provides the capability to query (by using SNMP commands) for routes that are learned from individual BGP peers.

Before this new MIB table was introduced, a network operator could obtain the routes learned by a local BGP-speaking router by querying the local BGP speaker with an SNMP command (for example, the `snmpwalk` command). The network operator used the SNMP command to query the bgp4PathAttrTable of the CISCO-BGP4-MIB. The routes that were returned from a bgp4PathAttrTable query were indexed in the following order:

- Prefix
- Prefix length
- Peer address

Because the bgp4PathAttrTable indexes the prefixes first, obtaining routes learned from individual BGP peers will require the network operator to "walk through" the complete bgp4PathAttrTable and filter out routes from the interested peer. A BGP Routing Information Base (RIB) could contain 10,000 or more routes, which makes a manual "walk" operation impossible and automated walk operations very inefficient.

BGP 4 MIB Support for per-Peer Received Routes introduces a Cisco-specific enterprise extension to the CISCO-BGP4-MIB that defines a new table called the cbgpRouterTable. The cbgpRouterTable provides the same information as the bgp4PathAttrTable with the following two differences:

- Routes are indexed in the following order:
  - Peer address
  - Prefix
  - Prefix length

The search criteria for SNMP queries of local routes are improved because peer addresses are indexed before prefixes. A search for routes that are learned from individual peers is improved with this enhancement because peer addresses are indexed before prefixes. A network operator will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.

- Support is added for multiprotocol BGP, Address Family Identifier (AFI), and Subsequent Address Family Identifier (SAFI) information. This information is added in the form of indexes to the cbgpRouterTable. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.
The MIB will be populated only if the router is configured to run a BGP process. The present implementation of BGP 4 MIB Support for Per-Peer Received Routes will show only routes contained in IPv4 AFI and unicast SAFI BGP local RIB tables. Support for showing routes contained in other local RIB tables will be added in the future.

BGP 4 Per-Peer Received Routes Table Elements and Objects

The following sections describe new table elements, AFI and SAFI tables and objects, and network address prefixes in the Network Layer Reachability Information (NLRI) fields that have been introduced by the BGP 4 MIB Support for Per-Peer Received Routes enhancement.

- MIB Tables and Objects, page 523
- AFIs and SAFIs, page 524
- Network Address Prefix Descriptions for the NLRI Field, page 524

MIB Tables and Objects

The table below describes the MIB indexes of the cbgpRouterTable.

For a complete description of the MIB, see the CISCO-BGP4-MIB file CISCO-BGP4-MIB.my, available through Cisco.com at the following URL:


<table>
<thead>
<tr>
<th>MIB Indexes of the cbgpRouterTable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIB Indexes</strong></td>
</tr>
<tr>
<td>cbgpRouteAfi</td>
</tr>
<tr>
<td>cbgpRouteSafi</td>
</tr>
<tr>
<td>cbgpRoutePeerType</td>
</tr>
<tr>
<td>cbgpRoutePeer</td>
</tr>
</tbody>
</table>
| cbgpRouteAddrPrefix              | Represents the network address prefix that is carried in a BGP update message.  
See the table below for information about the types of network layer addresses that can be stored in specific types of AFI and SAFI objects. |
MIB Indexes

<table>
<thead>
<tr>
<th>MIB Indexes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cbgpRouteAddrPrefixLen</td>
<td>Represents the length in bits of the network address prefix in the NLRI field.</td>
</tr>
<tr>
<td></td>
<td>See the table below for a description of the 13 possible entries.</td>
</tr>
</tbody>
</table>

AFIs and SAFIs

The table below lists the AFI and SAFI values that can be assigned to or held by the cbgpRouteAfi and cbgpRouteSafi indexes, respectively. The table below also displays the network address prefix type that can be held by specific combinations of AFIs and SAFIs. The type of network address prefix that can be carried in a BGP update message depends on the combination of AFIs and SAFIs.

Table 32  AFIs and SAFIs

<table>
<thead>
<tr>
<th>AFI</th>
<th>SAFI</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4(1)</td>
<td>unicast(1)</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>ipv4(1)</td>
<td>multicast(2)</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>ipv4(1)</td>
<td>vpn(128)</td>
<td>VPN-IPv4 address</td>
</tr>
<tr>
<td>ipv6(2)</td>
<td>unicast(1)</td>
<td>IPv6 address</td>
</tr>
</tbody>
</table>

Note

A VPN-IPv4 address is a 12-byte quantity that begins with an 8-byte Route Distinguisher (RD) and ends with a 4-byte IPv4 address. Any bits beyond the length specified by cbgpRouteAddrPrefixLen are represented as zeros.

Network Address Prefix Descriptions for the NLRI Field

The table below describes the length in bits of the network address prefix in the NLRI field of the cbgpRouteTable. Each entry in the table provides information about the route that is selected by any of the six indexes in the table below.

Table 33  Network Address Prefix Descriptions for the NLRI Field

<table>
<thead>
<tr>
<th>Table or Object (or Index)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cbgpRouteOrigin</td>
<td>The ultimate origin of the route information.</td>
</tr>
<tr>
<td>cbgpRouteASPathSegment</td>
<td>The sequence of autonomous system path segments.</td>
</tr>
<tr>
<td>cbgpRouteNextHop</td>
<td>The network layer address of the autonomous system border router that traffic should pass through to get to the destination network.</td>
</tr>
</tbody>
</table>
Table or Object (or Index) | Description
--- | ---
cbgpRouteMedPresent | Indicates that the MULTI_EXIT_DISC attribute for the route is either present or absent.
cbgpRouteMultiExitDisc | Metric that is used to discriminate between multiple exit points to an adjacent autonomous system. The value of this object is irrelevant if the value of the cbgpRouteMedPresent object is "false(2)."
cbgpRouteLocalPrefPresent | Indicates that the LOCAL_PREF attribute for the route is either present or absent.
cbgpRouteLocalPref | Determines the degree of preference for an advertised route by an originating BGP speaker. The value of this object is irrelevant if the value of the cbgRouteLocalPrefPresent object is "false(2)."
cbgpRouteAtomicAggregate | Determines if the system has selected a less specific route without selecting a more specific route.
cbgpRouteAggregatorAS | The autonomous system number of the last BGP speaker that performed route aggregation. A value of 0 indicates the absence of this attribute.
cbgpRouteAggregatorAddrType | Represents the type of network layer address that is stored in the cbgpRouteAggregatorAddr object.
cbgpRouteAggregatorAddr | The network layer address of the last BGP 4 speaker that performed route aggregation. A value of all zeros indicates the absence of this attribute.
cbgpRouteBest | An indication of whether this route was chosen as the best BGP 4 route.
cbgpRouteUnknownAttr | One or more path attributes not understood by the local BGP speaker. A size of 0 indicates that this attribute is absent.

**Benefits of BGP 4 MIB Support for Per-Peer Received Routes**

- Improved SNMP Query Capabilities--The search criteria for SNMP queries for routes that are advertised by individual peers are improved because the peer address is indexed before the prefix. A network operator will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.
- Improved AFI and SAFI Support--Support is added for multiprotocol BGP. AFI and SAFI are added as indexes to the table. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.
### Additional References

The following sections provide references related to BGP 4 MIB Support for Per-Peer Received Routes.

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring MIBs for BGP</td>
<td>&quot;Configuring Advanced BGP Features&quot;</td>
</tr>
<tr>
<td>BGP commands</td>
<td><em>Cisco IOS IP Routing: BGP Command Reference</em></td>
</tr>
<tr>
<td>Configuring SNMP Support</td>
<td>&quot; Configuring SNMP Support&quot;</td>
</tr>
<tr>
<td>SNMP commands</td>
<td><em>Cisco IOS Network Management Command Reference</em></td>
</tr>
<tr>
<td>Cisco IOS master command list, all releases</td>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1657</td>
<td><em>BGP-4 MIB</em></td>
</tr>
<tr>
<td>RFC 1771</td>
<td><em>A Border Gateway Protocol 4 (BGP-4)</em></td>
</tr>
<tr>
<td>RFC 2547</td>
<td><em>BGP/MPLS VPNs</em></td>
</tr>
<tr>
<td>RFC 2858</td>
<td><em>Multiprotocol Extensions for BGP-4</em></td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for BGP 4 MIB Support for Per-Peer Received Routes

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 34 Feature Information for BGP 4 MIB Support for Per-Peer Received Routes

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP 4 MIB support for per-peer received routes</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>BGP received routes MIB</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Glossary

AFI--Address Family Identifier. Carries the identity of the network layer protocol that is associated with the network address.

BGP--Border Gateway Protocol. An interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163, A Border Gateway Protocol (BGP). The current implementation of BGP is BGP Version 4 (BGP4). BGP4 is the predominant interdomain routing protocol
that is used on the Internet. It supports CIDR and uses route aggregation mechanisms to reduce the size of routing tables.

MBGP--multiprotocol BGP. An enhanced version of BGP that carries routing information for multiple network layer protocols and IP multicast routes. It is defined in RFC 2858, Multiprotocol Extensions for BGP-4.

MIB--Management Information Base. A group of managed objects that are contained within a virtual information store or database. MIB objects are stored so that values can be assigned to object identifiers and to assist managed agents by defining which MIB objects should be implemented. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

NLRI--Network Layer Reachability Information. Carries route attributes that describe a route and how to connect to a destination. This information is carried in BGP update messages. A BGP update message can carry one or more NLRI prefixes.

RIB--Routing Information Base (RIB). A central repository of routes that contains Layer 3 reachability information and destination IP addresses or prefixes. The RIB is also known as the routing table.

SAFI--Subsequent Address Family Identifier. Provides additional information about the type of the Network Layer Reachability Information that is carried in the attribute.


**snmpwalk** --The snmpwalk command is an SNMP application that is used to communicate with a network entity MIB using SNMP.

VPN--Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses a tunnel to encrypt all information at the IP level.
BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced.

- Finding Feature Information, page 529
- Prerequisites for BGP Event-Based VPN Import, page 529
- Information About BGP Event-Based VPN Import, page 529
- How to Configure BGP Event-Based VPN Import, page 531
- Configuration Examples for BGP Event-Based VPN Import, page 537
- Where to Go Next, page 538
- Additional References, page 538
- Feature Information for BGP Event-Based VPN Import, page 539

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP Event-Based VPN Import

Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Information About BGP Event-Based VPN Import

- BGP Event-Based VPN Import, page 530
BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing BGP path import process. BGP Virtual Private Network (VPN) import provides importing functionality for BGP paths where BGP paths are imported from the BGP VPN table into a BGP virtual routing and forwarding (VRF) topology. In the existing path import process, when path updates occur, the import updates are processed during the next scan time which is a configurable interval of 5 to 15 seconds. The scan time adds a delay in the propagation of routes. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available.

Using the BGP Event-Based VPN Import feature, convergence times are significantly reduced because provider edge (PE) routers can propagate VPN paths to customer edge (CE) routers without the scan time delay. Configuration changes such as adding imported route-targets to a VRF are not processed immediately, and are still handled during the 60-second periodic scanner pass.

- Import Path Selection Policy, page 530
- Import Path Limit, page 530

Import Path Selection Policy

Event-based VPN import introduces three path selection policies:

- **All**—Import all available paths from the exporting net that match any route target (RT) associated with the importing VRF instance.
- **Best path**—Import the best available path that matches the RT of the VRF instance. If the best path in the exporting net does not match the RT of the VRF instance, a best available path that matches the RT of the VRF instance is imported.
- **Multipath**—Import the best path and all paths marked as multipaths that match the RT of the VRF instance. If there are no best path or multipath matches, then the best available path is selected.

Multipath and best path options can be restricted using an optional keyword to ensure that the selection is made only on the configured option. If the **strict** keyword is configured in the **import path selection** command, the software disables the fall back safety option of choosing the best available path. If no paths appropriate to the configured option (best path or multipath) in the exporting net match the RT of the VRF instance, then no paths are imported. This behavior matches the behavior of the software before the BGP Event-Based VPN Import feature was introduced.

When the restriction is not set, paths that are imported as the best available path are tagged. In **show** command output these paths are identified with the wording, "imported safety path."

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the **show** command output. Any path that is marked as "not-in-vrf" is not considered as a best path because paths not in the VRF appear less attractive than paths in the VRF.

Import Path Limit

To control the memory utilization, a maximum limit of the number of paths imported from an exporting net can be specified per importing net. When a selection is made of paths to be imported from one or more exporting net, the first selection priority is a best path, the next selection priority is for multipaths, and the lowest selection priority is for nonmultipaths.
How to Configure BGP Event-Based VPN Import

- Configuring a Multiprotocol VRF, page 531
- Configuring Event-Based VPN Import Processing for BGP Paths, page 534
- Monitoring and Troubleshooting BGP Event-Based VPN Import Processing, page 535

Configuring a Multiprotocol VRF

Perform this task to configure a multiprotocol VRF that allows you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. In this task, only the IPv4 address family is configured, but we recommend using the multiprotocol VRF configuration for all new VRF configurations.

Note

This task is not specific to the BGP Event-Based VPN Import feature.

SUMMARY STEPS

1. enable
2. configure terminal
3. vrf definition vrf-name
4. rd route-distinguisher
5. route-target {import | export | both} route-target-ext-community
6. address-family ipv4 [unicast]
7. exit-address-family
8. exit
9. interface type number
10. vrf forwarding vrf-name
11. ip address ip-address mask
12. no shutdown
13. exit
14. Repeat Step 3 through Step 13 to bind other VRF instances with an interface.
15. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

Example:

Router> enable
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vrf definition vrf-name</td>
<td>Configures a VRF routing table and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# vrf definition vrf-A</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> rd route-distinguisher</td>
<td>Creates routing and forwarding tables and specifies the default route distinguisher for a VPN.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# rd 45000:1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# route-target both 45000:100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> address-family ipv4 [unicast]</td>
<td>Specifies the IPv4 address family and enters VRF address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit-address-family</td>
<td>Exits VRF address family configuration mode and returns to VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf-af)# exit-address-family</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Exits VRF configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)# exit</td>
</tr>
<tr>
<td><strong>Step 9</strong> interface type number</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# interface FastEthernet 1/1</td>
</tr>
<tr>
<td><strong>Step 10</strong> vrf forwarding vrf-name</td>
<td>Associates a VRF instance with the interface configured in Step 9.</td>
</tr>
<tr>
<td></td>
<td>• When the interface is bound to a VRF, previously configured IP</td>
</tr>
<tr>
<td></td>
<td>addresses are removed, and the interface is disabled.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if)# vrf forwarding vrf-A</td>
</tr>
<tr>
<td><strong>Step 11</strong> ip address ip-address mask</td>
<td>Configures an IP address for the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if)# ip address 10.4.8.149 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 12</strong> no shutdown</td>
<td>Restarts a disabled interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if)# no shutdown</td>
</tr>
<tr>
<td><strong>Step 13</strong> exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 14</strong> Repeat Step 3 through Step 13 to bind other VRF instances with an interface</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 15</strong> end</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# end</td>
</tr>
</tbody>
</table>
How to Configure BGP Event-Based VPN Import

Configuring Event-Based VPN Import Processing for BGP Paths

Perform this task to reduce convergence times when BGP paths change by configuring event-based processing for importing BGP paths into a VRF table. Two new CLI commands allow the configuration of a maximum number of import paths per importing net and the configuration of a path selection policy.

This task assumes that you have previously configured the VRF to be used with the VRF address family syntax. To configure a VRF, see the Configuring a Multiprotocol VRF, page 531.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the Configuring Event-Based VPN Import Processing for BGP Paths Example, page 537.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 vrf vrf-name
5. import path selection {all | bestpath [strict] | multipath [strict]}
6. import path limit number-of-import-paths
7. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>address-family ipv4 vrf vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router(config-router)# address-family ipv4 vrf vrf-A</td>
</tr>
<tr>
<td>Specifies the IPv4 address family and enters address family configuration mode.&lt;br&gt;• Use the <code>vrf</code> keyword and <code>vrf-name</code> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**import path selection {all</td>
</tr>
<tr>
<td>Specifies the BGP path selection policy for importing routes into a VRF table.&lt;br&gt;• In this example, all paths that match any RT of the VRF instance are imported.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>import path limit number-of-import-paths</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router(config-router-af)# import path limit 3</td>
</tr>
<tr>
<td>Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>end</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router(config-router-af)# end</td>
</tr>
<tr>
<td>Exits address family configuration mode and returns to privileged EXEC mode.</td>
<td></td>
</tr>
</tbody>
</table>

### Monitoring and Troubleshooting BGP Event-Based VPN Import Processing

Perform the steps in this task as required to monitor and troubleshoot the BGP event-based VPN import processing.

Only partial command syntax for the show commands used in this task is displayed. For more details, see the Cisco IOS IP Routing: BGP Command Reference.

**SUMMARY STEPS**

1. **enable**
2. **show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [network-address [mask]]**
3. **show ip route [vrf vrf-name] [ip-address [mask]]**
4. **debug ip bgp vpnv4 unicast import {events | updates [access-list]}**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong>&lt;br&gt;Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
</tbody>
</table>
Example:

```
Router> enable
```

**Step 2**

```
show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [network-address [mask]]
```

In this example output, a safe import path selection policy is in effect because the `strict` keyword is not configured using the `import path selection` command. When a path is imported as the best available path (when the bestpath or multipaths are not eligible for import), the path is marked with "imported safety path,” as shown in the output.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0
```

```
BGP routing table entry for 45000:1:172.17.0.0/16, version 10
Paths: (1 available, best #1, table vrf-A)
Flag: 0x820
  Not advertised to any peer
  2, imported safety path from 50000:2:172.17.0.0/16
    10.0.101.1 from 10.0.101.1 (10.0.101.1)
    Origin IGP, metric 200, localpref 100, valid, internal, best
    Extended Community: RT:45000:100
```

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf” in the `show` command output.

In the following example output, a path was received from another peer router and was not subject to the VPN importing rules. This path, 10.0.101.2, was added to the VPNv4 table and associated with the vrf-A net because it contains a match of the RD information although the RD information was from the original router. This path is not, however, an RT match for vrf-A and is marked as "not-in-vrf.” Note that on the net for vrf-A, this path is not the bestpath because any paths that are not in the VRF appear less attractive than paths in the VRF.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0
```

```
BGP routing table entry for 45000:1:172.17.0.0/16, version 11
Paths: (2 available, best #2, table vrf-A)
Flag: 0x820
  Not advertised to any peer
  2, imported safety path from 50000:2:172.17.0.0/16
    10.0.101.2 from 10.0.101.2 (10.0.101.2)
    Origin IGP, metric 100, localpref 100, valid, internal, not-in-vrf
    Extended Community: RT:45000:200
    mpls labels in/out nolabel/16
    2, imported safety path from 50000:2:172.17.0.0/16
      10.0.101.1 from 10.0.101.1 (10.0.101.1)
      Origin IGP, metric 50, localpref 100, valid, internal, best
      Extended Community: RT:45000:100
      mpls labels in/out nolabel/16
```

**Step 3**

```
show ip route [vrf vrf-name] [ip-address [mask]]
```

In this example output, information about the routing table for VRF vrf-A is displayed:

Example:

```
Router# show ip route vrf vrf-A 172.17.0.0
```

```
Routing Table: vrf-A
```
Routing entry for 172.17.0.0/16
Known via "bgp 1", distance 200, metric 50
Tag 2, type internal
Last update from 10.0.101.33 00:00:32 ago
Routing Descriptor Blocks:
* 10.0.101.33 (default), from 10.0.101.33, 00:00:32 ago
  Route metric is 50, traffic share count is 1
  AS Hops 1
  Route tag 2
  MPLS label: 16
  MPLS Flags: MPLS Required

Step 4  
**debug ip bgp vpnv4 unicast import {events | updates [access-list]}**

Use this command to display debugging information related to the importing of BGP paths into a VRF instance table. The actual output depends on the commands that are subsequently entered.

Note: If no access list to filter prefixes is specified when using the updates keyword, all updates for all prefixes are displayed and this may slow down your network.

Example:

Router# debug ip bgp vpnv4 unicast import events
BGP import events debugging is on

---

**Configuration Examples for BGP Event-Based VPN Import**

-  [Configuring Event-Based VPN Import Processing for BGP Paths Example, page 537](IP Routing: BGP Configuration Guide, Cisco IOS XE Release 2)

**Configuring Event-Based VPN Import Processing for BGP Paths Example**

In this example configuration, a VRF (vrf-A) is configured and VRF forwarding is applied to Fast Ethernet interface 1/1. In address family mode the import path selection is set to all and the number of import paths is set to 3. Two BGP neighbors are configured under the IPv4 address family and activated under the VPNv4 address family.

```plaintext
vrf definition vrf-A
  rd 45000:1
  route-target import 45000:100
  address-family ipv4
  exit-address-family
!
interface FastEthernet1/1
  no ip address
  vrf forwarding vrf-A
  ip address 10.4.8.149 255.255.255.0
  no shut
  exit
!
router bgp 45000
  network 172.17.1.0 mask 255.255.255.0
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.1.3 remote-as 50000
  address-family ipv4
    vrf vrf-A
    import path selection all
    import path limit 3
```
exit-address-family
address-family vpnv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
end

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" module.
- If you want to configure BGP neighbor session options, see the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some advanced BGP features, see the "Configuring Advanced BGP Features" module.

Additional References

The following sections provide references related to the BGP Event-Based VPN Import feature.

<table>
<thead>
<tr>
<th>Related Documents</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Topic</td>
<td>Document Title</td>
</tr>
<tr>
<td>BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Overview of Cisco BGP conceptual information with links to all the individual BGP modules</td>
<td>&quot;Cisco BGP Overview&quot; module of the Cisco IOS IP Routing: BGP Configuration Guide.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Lookup Tool</th>
<th><a href="http://tools.cisco.com/Support/CLILookup">http://tools.cisco.com/Support/CLILookup</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Title</td>
</tr>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>

Feature Information for BGP Event-Based VPN Import

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 35  Feature Information for BGP Event-Based VPN Import

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Event-Based VPN Import</td>
<td>Cisco IOS XE Release 2.6</td>
<td>The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced. The following commands were introduced or modified:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>bgp scan-time</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>import path limit</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>import path selection</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>maximum-path ebgp</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>maximum-path ibgp</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>show ip bgp vpnv4</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>show ip bgp vpnv6</code></td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
BGP Best External

The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. The BGP Best External feature advertises the most preferred route among those received from external neighbors as a backup route. This feature is beneficial in active-backup topologies, where service providers use routing policies that cause a border router to choose a path received over an internal BGP (iBGP) session (of another border router) as the best path for a prefix even if it has an external BGP (eBGP) learned path. This active-backup topology defines one exit or egress point for the prefix in the autonomous system and uses the other points as backups if the primary link or eBGP peering is unavailable. The policy causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because it does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path.

- Finding Feature Information, page 541
- Prerequisites for BGP Best External, page 541
- Restrictions for BGP Best External, page 542
- Information About BGP Best External, page 542
- How to Configure BGP Best External, page 546
- Configuration Examples for BGP Best External, page 555
- Additional References, page 556
- Feature Information for BGP Best External, page 557

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP Best External

- The Bidirectional Forwarding Detection (BFD) protocol must be enabled to quickly detect link failures.
- Ensure that the BGP and the Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- The backup path must have a unique next hop that is not the same as the next hop of the best path.
- BGP must support lossless switchover between operational paths.

**Restrictions for BGP Best External**

- The BGP Best External feature will not install a backup path if BGP Multipath is installed and a multipath exists in the BGP table. One of the multipaths automatically acts as a backup for the other paths.
- The BGP Best External feature is not supported with the following features:
  - MPLS VPN Carrier Supporting Carrier
  - MPLS VPN Inter-Autonomous Systems, option B
  - MPLS VPN Per Virtual Routing and Forwarding (VRF) Label
- The BGP Best External feature cannot be configured with Multicast or L2VPN VRF address families.
- The BGP Best External feature cannot be configured on a route reflector, unless it is running Cisco IOS XE Release 3.4S or later.
- The BGP Best External feature does not support NSF/SSO. However, ISSU is supported if both Route Processors have the BGP Best External feature configured.
- The BGP Best External feature can only be configured on VPNv4, VPNv6, IPv4 VRF, and IPv6 VRF address families.
- When you configure the BGP Best External feature using the `bgp advertise-best-external` command, you need not enable the BGP PIC feature with the `bgp additional-paths install` command. The BGP PIC feature is automatically enabled by the BGP Best External feature.
- When you configure the BGP Best External feature, it will override the functionality of the "MPLS VPN--BGP Local Convergence" feature. However, you do not have to remove the `protection local-prefixes` command from the configuration.

**Information About BGP Best External**

- BGP Best External Overview, page 542
- What the Best External Route Means, page 543
- How the BGP Best External Feature Works, page 543
- Configuration Modes for Enabling BGP Best External, page 544
- BGP Best External Path on RR for Intercluster, page 544
- CLI Differences for Best External Path on an RR for Intercluster, page 545
- Rules Used to Calculate the BGP Best External Path for Intercluster RRs, page 545

**BGP Best External Overview**

Service providers use routing policies that cause a border router to choose a path received over an iBGP session (of another border router) as the best path for a prefix even if it has an eBGP learned path. This practice is popularly known as active-backup topology and is done to define one exit or egress point for the prefix in the autonomous system and to use the other points as backups if the primary link or eBGP peering is unavailable.

The policy, though beneficial, causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because the border router does not advertise any path for such prefixes. To
cope with this situation, some routers advertise one externally learned path called the best external path. The best external behavior causes the BGP selection process to select two paths to every destination:

- The best path is selected from the complete set of routes known to that destination.
- The best external path is selected from the set of routes received from its external peers.

BGP advertises the best path to external peers. Instead of withdrawing the best path from its internal peers when it selects an iBGP path as the best path, BGP advertises the best external path to the internal peers.

The BGP Best External feature is an essential component of the Prefix-Independent Convergence (PIC) edge for both Internet access and MPLS VPN scenarios and makes alternate paths available in the network in the active-backup topology.

What the Best External Route Means

The BGP Best External feature uses a "best external route" as a backup path, which, according to draft-marques-idr-best-external, is the most preferred route among those received from external neighbors. The most preferred route from external neighbors can be the following:

- Two routers in different clusters that have an iBGP session between them.
- Two routers in different autonomous systems of a confederation that have an eBGP session between them.

The best external route might be different from the best route installed in the routing information base (RIB). The best route could be an internal route. By allowing the best external route to be advertised and stored, in addition to the best route, networks gain faster restoration of connectivity by providing additional paths that may be used if the primary path fails.

How the BGP Best External Feature Works

The BGP Best External feature is based on Internet Engineering Task Force (IETF) draft-marques-idr-best-external.txt. The BGP Best External feature advertises a best external route to its internal peers as a backup route. The backup route is stored in the RIB and Cisco Express Forwarding. If the primary path fails, the BGP PIC functionality enables the best external path to take over, enabling faster restoration of connectivity.

![MPLS VPN: Best External at the Edge of MPLS VPN](image)
The figure above shows an MPLS VPN using the BGP Best External feature. The network includes the following components:

- eBGP sessions exist between the provider edge (PE) and customer edge (CE) routers.
- PE1 is the primary router and has a higher local preference setting.
- Traffic from CE2 uses PE1 to reach router CE1.
- PE1 has two paths to reach CE1.
- CE1 is dual-homed with PE1 and PE2.
- PE1 is the primary path and PE2 is the backup path.

In the figure above, traffic in the MPLS cloud flows through PE1 to reach CE1. Therefore, PE2 uses PE1 as the best path and PE2 as the backup path.

PE1 and PE2 are configured with the BGP Best External feature. BGP computes both the best path (the PE1-CE1 link) and a backup path (PE2) and installs both paths into the RIB and Cisco Express Forwarding. The best external path (PE2) is advertised to the peer routers, in addition to the best path.

When Cisco Express Forwarding detects a link failure on the PE1-CE1 link, Cisco Express Forwarding immediately switches to the backup path PE2. Traffic is quickly rerouted due to local Fast Convergence in Cisco Express Forwarding using the backup path. Thus, traffic loss is minimized and fast convergence is achieved.

**Configuration Modes for Enabling BGP Best External**

You can enable the BGP Best External feature in different modes, each of which protects VRFs in its own way:

- If you issue the `bgp advertise-best-external` command in VPNv4 address family configuration mode, it applies to all IPv4 VRFs. If you issue the command in this mode, you need not issue it for specific VRFs.
- If you issue the `bgp advertise-best-external` command in IPv4 address family configuration mode, it applies only that VRF.

**BGP Best External Path on RR for Intercluster**

Beginning with Cisco IOS XE Release 3.4S, BGP Best External is extended to BGP Best External for Intercluster RRs. This feature provides path diversity between RR clusters, providing best external functionality toward non-client iBGP peers. The feature is also known as the “intercluster best external path.”

Best external path at an RR means the best path within the RR’s cluster. This path might also be referred to as the best internal path.

When an RR (RR1) chooses a non-client iBGP path (that is, a path learned from another RR, let’s say RR2) as its overall best, with the BGP Best External for Intercluster RRs feature, RR1 will be able to advertise its best internal path to the non-client iBGP peers. This will help RR2 to learn an additional path, providing a diverse path.

Best external functionality at RRs is only for non-client iBGP peers. An RR cannot advertise best external paths to its clients because it has to advertise its overall bestpath (which can be either a client path or non-client or eBGP path).

The best external path calculated by the RR is the best internal path for the cluster. It will be advertised to the non-client iBGP peers only when the overall best path at this RR is a non-client iBGP path.
When there are multiple RRs, each in its own cluster, each RR must have the `neighbor advertise best-external` command configured for each of its neighbor RRs.

If the RR is in the forwarding plane, the `bgp additional paths install` command is necessary.

## CLI Differences for Best External Path on an RR for Intercluster

Prior to Cisco IOS XE Release 3.4S, the BGP Best External feature was allowed on a PE only, and it was configured by the `bgp advertise-best-external` command. The calculation of the backup path, installation, and advertisement were tied together in one command.

Beginning with Cisco IOS XE Release 3.4S, the BGP Best External feature is allowed on PEs and RRs. The functionality of the `bgp advertise-best-external` command is divided among the following three commands that calculate, install, and advertise the best external path:

- `bgp additional-path select best-external`
- `bgp additional-path install`
- `neighbor advertise diverse-path best-external`

If the `bgp additional-path select best-external` command is not configured, the system will calculate and install the best external path, but not advertise it.

The `neighbor advertise diverse-path best-external` command enables the advertisement of the best external path to the specified neighbor.

## Rules Used to Calculate the BGP Best External Path for Intercluster RRs

The best internal path implementation on an RR toward non-clients (different cluster RRs) is calculated based on the following rules:

1. Calculate the overall primary bestpath on the RR per the normal bestpath selection rules.
2. If a backup path configuration is enabled, calculate the second bestpath (which is a different path from the primary bestpath selected in Rule 1 and has a different nexthop from this bestpath), which is marked as the backup path. Backup path selection is enabled using the `bgp additional-paths install` or `bgp additional-paths select [best-external] [backup]` command.
3. If the overall best path on the RR is a non-client iBGP path and not an eBGP path, calculate the best external/internal path from the remaining paths after excluding results from Rule 1 and Rule 2 and by ignoring all the other paths from the other clusters and run normal bestpath rules by including all the remaining eBGP and iBGP paths. Select the newly obtained bestpath and mark it as the best internal path.
4. Advertise this best internal path, which is either eBGP (received from CE peers for RR/ASBR) or iBGP (received from RR clients) toward the non-client RRs when `neighbor advertise best-external` is configured towards the non-client RRs.
5. If the overall bestpath is a path received from either an RR client or eBGP peer (in case of RR/ASBR) either an iBGP or an eBGP path will be chosen as bestpath per the normal bestpath algorithm. Because the overall bestpath is an internal client path, the normal advertisement rules will automatically advertise this path to non-client iBGP peers/RRs. This behavior is the same as the existing behavior (when best external is not enabled on RRs) when an RR client’s path is chosen as the overall bestpath.
6. We do not allow a best external path to be configured on an RR towards RR-clients. The `neighbor advertise best-external` command can be configured on RR/ASBR only for non-clients or peering with RRs in the other clusters.
7. When multipath is enabled on the RR and only when the overall bestpath is from a non-client and if some of the intracluster client paths are also marked as multipaths, when best external is enabled on the RR (`neighbor advertise best-external` towards the RR non-client), the algorithm selects the older
multipath among the intra-cluster client multipaths (paths obtained from RR clients and eBGP peers within the cluster) and marks it as best internal path and announces it to the non-clients as best external path, so that the non-clients get path diversity from this cluster. If there are no intra-cluster multipaths found, we choose the best external path per Rules 3 through 5.

How to Configure BGP Best External

- Enabling the BGP Best External Feature, page 546
- Verifying the BGP Best External Feature, page 548
- Configuring Best External Path on an RR for an Intercluster, page 551

Enabling the BGP Best External Feature

Perform the following task to enable the BGP Best External feature. This task shows how to configure the BGP Best External feature in either IPv4 or VPNv4 address family. In VPNv4 address family configuration mode, the BGP Best External feature applies to all IPv4 VRFs; you do not have to configure it for specific VRFs. If you issue the `bgp advertise-best-external` command in IPv4 VRF address family configuration mode, the BGP Best External feature applies only that VRF.

- Configure the MPLS VPN and verify that it is working properly before configuring the BGP Best External feature. See "Configuring MPLS Layer 3 VPNs" for more information.
- Configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs".
- Ensure that the CE router is connected to the network by at least two paths.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. Do one of the following:
   - `address-family ipv4 [unicast | vrf vrf-name]`
   - or
   - `address-family vpnv4 [unicast]`
5. bgp advertise-best-external
6. neighbor ip-address remote-as autonomous-system-number
7. neighbor ip-address activate
8. neighbor ip-address fall-over [b fd]route-map map-name]
9. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. |
| **Example:** Router(config)# router bgp 40000 |
| **Step 4** Do one of the following:  
  - address-family ipv4 [unicast | vrf vrf-name]  
  - or  
  - address-family vpnv4 [unicast] | Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.  
- The **unicast** keyword specifies the IPv4 or VPNv4 unicast address family.  
- The **vrf** keyword and **vrf-name** argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. |
| **Example:** Router(config-router)# address-family ipv4 unicast |
| **Example:** Router(config-router)# address-family vpnv4 |
| **Step 5** bgp advertise-best-external | Calculates and uses an external backup path and installs it into the RIB and Cisco Express Forwarding. |
| **Example:** Router(config-router-af)# bgp advertise-best-external |
### Command or Action | Purpose
---|---
**Step 6** neighbor ip-address remote-as autonomous-system-number | Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.  
- By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types.

Example:

```
Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000
```

**Step 7** neighbor ip-address activate | Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.

Example:

```
Router(config-router-af)# neighbor 192.168.1.1 activate
```

**Step 8** neighbor ip-address fall-over [bfd | route-map map-name] | Configures the BGP peering to use fast session deactivation and enables BFD protocol support for failover.  
- BGP will remove all routes learned through this peer if the session is deactivated.

Example:

```
Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd
```

**Step 9** end | (Optional) Exits address family configuration mode and returns to privileged EXEC mode.

Example:

```
Router(config-router-af)# end
```

### Verifying the BGP Best External Feature

Perform the following task to verify that the BGP Best External feature is configured correctly.

**SUMMARY STEPS**

1. enable
2. show vrf detail
3. show ip bgp ipv4 mdt all | rd vrf | multicast | tunnel unicast or show ip bgp vpn4 all rd route-distinguisher | vrf vrf-name rib-failure ip-prefix/length longer-prefixes | network-address mask longer-prefixes | cidr-only community community-list dampened-paths filter-list | [flap-statistics inconsistent-as neighbors paths line] | peer-group quote-regexp regexp | [summary labels
4. show bpg vpnv4 unicast vrf vrf-name ip-address
5. show ip route vrf vrf-name repair-paths ip-address
6. show ip cef vrf vrf-name ip-address detail
DETAILED STEPS

**Step 1**
*enable*

Use this command to enable privileged EXEC mode. Enter your password, if prompted. For example:

**Example:**
```
Router> enable
Router#
```

**Step 2**
*show vrf detail*

Use this command to verify that the BGP Best External feature is enabled. The following `show vrf detail` command output shows that the BGP Best External feature is enabled.

**Example:**
```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
   Interfaces:
      Se4/0
   Address family ipv4 (Table ID = 1 {0x1}):
      Export VPN route-target communities
         RT:100:1                 RT:200:1                 RT:300:1
         RT:400:1
      Import VPN route-target communities
         RT:100:1                 RT:200:1                 RT:300:1
         RT:400:1
   No import route-map
   No export route-map
   VRF label distribution protocol: not configured
   VRF label allocation mode: per-prefix
   Prefix protection with additional path enabled
   Address family ipv6 not active.
```

**Step 3**
*show ip bgp ipv4 mdt all | rd vrf | multicast | tunnel unicast or show ip bgp vpn4 all rd route-distinguisher | vrf vrf-name rib-failure ip-prefix/length longer-prefixes] [network-address mask longer-prefixes] [cidr-only community community-list dampened-paths filter-list] [flap-statistics inconsistent-as neighbors paths line] peer-group quote-regexp regexp] [summary labels]*

Use this command to verify that the best external route is advertised. In the command output, the code b indicates a backup path and the code x designates the best external path.

**Example:**
```
Router# show ip bgp vpnv4 all
BGP table version is 1104964, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale, multipath,
 b backup-path, x best-external
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network   Next Hop   Metric  LocPrf  Weight  Path
Route Distinguisher: 11:12 (default for vrf blue)
  *>11.0.0.1/32 10.10.3.3 0 200 0 1 ?
  * i 10.10.3.3 0 200 0 1 ?
  * 10.0.0.1 0 0 1 ?
  *bx 10.0.0.1 0 0 1 ?
  * 10.0.0.1 0 0 1 ?
```

**Step 4**
*show bgp vpnv4 unicast vrf vrf-name ip-address*

```
```
Use this command to verify that the best external route is advertised.

**Example:**

Router# show bgp vpnv4 unicast vrf vpn1 10.10.10.10
BGP routing table entry for 10.10.10.10.10.10.10.10/32, version 10
Paths: (2 available, best #1, table vpn1)
  Advertise-best-external
  Advertised to update-groups:
    1 2
    200
    10.6.6.6 (metric 21) from 10.6.6.6 (10.6.6.6)
    Origin incomplete, metric 0, localpref 200, valid, internal, best
    Extended Community: RT:1:1
    mpls labels in/out 23/23
    200
    10.1.2.1 from 10.1.2.1 (10.1.1.1)
    Origin incomplete, metric 0, localpref 100, valid,
    external, backup/repair, advertise-best-external
    Extended Community: RT:1:1, recursive-via-connected
    mpls labels in/out 23/nolabel

**Step 5**

`show ip route vrf vrf-name repair-paths ip-address`
Use this command to display the repair route.

**Example:**

Router# show ip route vrf vpn1 repair-paths

Routing Table: vpn1
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
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
* - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP
+ - replicated route, % - next hop override
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B 10.1.1.0/24 [200/0] via 10.6.6.6, 00:38:33
    [RPR](200/0) via 10.1.2.1, 00:38:33
B 10.1.1.1/32 [200/0] via 10.6.6.6, 00:38:33
    [RPR](200/0) via 10.1.2.1, 00:38:33
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.1.2.0/24 is directly connected, Ethernet0/0
L 10.1.2.2/32 is directly connected, Ethernet0/0
B 10.1.6.0/24 [200/0] via 10.6.6.6, 00:38:33
    [RPR](200/0) via 10.1.2.1, 00:38:33

**Step 6**

`show ip cef vrf vrf-name ip-address detail`
Use this command to display the best external route.

**Example:**

Router# show ip cef vrf test 10.71.8.164 detail
10.71.8.164/30, epoch 0, flags rib defined all labels
    recursive via 10.249.0.102 label 35
    nexthop 10.249.246.101 Ethernet0/0 label 25
    recursive via 10.249.0.104 label 28,
    repair
    nexthop 10.249.246.101 Ethernet0/0 label 24
Configuring Best External Path on an RR for an Intercluster

Perform the following task to configure a best external path on an RR for an intercluster. The steps in this particular task configure RR1 in the figure below, in the IPv4 address family. The step that configures address family lists the other address families supported.

Figure 48 Scenario for Configuring a BGP Best External Path on a RR for an Intercluster
### SUMMARY STEPS

1. enable
2. configure terminal
3. **router bgp** autonomous-system-number
4. **neighbor** *ip-address* remote-as autonomous-system-number
5. **neighbor** *ip-address* remote-as autonomous-system-number
6. address-family ipv4 unicast
7. **neighbor** *ip-address* activate
8. **neighbor** *ip-address* activate
9. bgp additional-paths select best-external
10. bgp additional-paths install
11. neighbor *ip-address* advertise best-external
12. neighbor *ip-address* advertise best-external
13. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Example:** | 
- Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** | 
- Router# configure terminal |
| **Step 3** **router bgp** autonomous-system-number | Enters router configuration mode for the specified routing process. |
| **Example:** | 
- Router(config)# router bgp 1 |
| **Step 4** **neighbor** *ip-address* remote-as autonomous-system-number | Adds an entry to the BGP or multiprotocol BGP neighbor table.  
- This step is for RR3. |
| **Example:** | 
- Router(config-router)# neighbor 10.5.1.1 remote-as 1 |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 5** neighbor ip-address remote-as autonomous-system-number | Adds an entry to the BGP or multiprotocol BGP neighbor table.  
- This step is for RR6. |
| **Example:**  
Router(config-router)# neighbor 10.5.1.2  
remote-as 1 | |
| **Step 6** address-family ipv4 unicast | Specifies the address family and enters address family configuration mode.  
- Supported address families are ipv4 unicast, vpnv4 unicast, ipv6 unicast, vpnv6 unicast, ipv4+label, and ipv6+label. |
| **Example:**  
Router(config-router)# address-family ipv4 unicast | |
| **Step 7** neighbor ip-address activate | Enables the exchange of information with a BGP neighbor.  
- This step is for RR3. |
| **Example:**  
Router(config-router-af)# neighbor 10.5.1.1 activate | |
| **Step 8** neighbor ip-address activate | Enables the exchange of information with a BGP neighbor.  
- This step is for RR6. |
| **Example:**  
Router(config-router-af)# neighbor 10.5.1.2 activate | |
| **Step 9** bgp additional-paths select best-external | Configures the system to calculate a best external path (external to RR cluster). |
| **Example:**  
Router(config-router-af)# bgp additional-paths select best-external | |
| **Step 10** bgp additional-paths install | Enables BGP to calculate a backup path for a given address family and to install it into the RIB and CEF.  
- This step is necessary if the RR is enabled for forwarding (the RR is in the forwarding plane). Otherwise, this step is unnecessary. |
| **Example:**  
Router(config-router-af)# bgp additional-paths install | |
| **Step 11** neighbor ip-address advertise best-external | (Optional) Configures a neighbor to receive the best external path in an advertisement.  
- This step is for RR3. |
| **Example:**  
Router(config-router-af)# neighbor 10.5.1.1 advertise best-external | |
### Command or Action

**Step 12** *neighbor ip-address advertise best-external*

*Purpose*

(Optional) Configures a neighbor to receive the best external path in an advertisement.

- This step is for RR6.

**Example:**

```
Router(config-router-af)# neighbor 10.5.1.2 advertise best-external
```

**Step 13** *end*

*Purpose*

(Optional) Exits address family configuration mode and returns to privileged EXEC mode.

**Example:**

```
Router(config-router-af)# end
```

In the scenario shown above, the following paths are selected as best path, backup path, and best internal path on the three RRs located in the three different clusters:

**On RR1:**

**On RR3:**

**On RR6:**

<table>
<thead>
<tr>
<th>To Reach Prefix 10/8</th>
<th>Next Hop:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PE5 (best path, local preference = 200)</td>
</tr>
<tr>
<td></td>
<td>PE3 (backup path, local preference = 150)</td>
</tr>
<tr>
<td></td>
<td>PE3 (best internal path, local preference = 150)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To Reach Prefix 10/8</th>
<th>Next Hop:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PE5 (best path, local preference = 200)</td>
</tr>
<tr>
<td></td>
<td>PE6 (backup path, local preference = 50)</td>
</tr>
<tr>
<td></td>
<td>PE3 (received as best external path from RR1, local preference = 150)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To Reach Prefix 10/8</th>
<th>Next Hop:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PE5 (best path, local preference = 200)</td>
</tr>
<tr>
<td></td>
<td>PE6 (backup path, local preference = 50)</td>
</tr>
<tr>
<td></td>
<td>PE3 (received as best external path from RR1, local preference = 150)</td>
</tr>
</tbody>
</table>
Configuration Examples for BGP Best External

- Example Configuring the BGP Best External Feature, page 555
- Example Configuring a Best External Path on an RR for an Intercluster, page 555

Example Configuring the BGP Best External Feature

The following example shows how to configure the BGP Best External feature in VPNv4 mode:

```plaintext
vrf definition test1
  rd 400:1
  route-target export 100:1
  route-target export 200:1
  route-target export 300:1
  route-target export 400:1
  route-target import 100:1
  route-target import 200:1
  route-target import 300:1
  route-target import 400:1
address-family ipv4
  exit-address-family
  exit
! interface Ethernet1/0
  vrf forwarding test1
  ip address 10.0.0.1 255.0.0.0
  exit
!
router bgp 64500
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.5.5.5 remote-as 64500
  neighbor 10.5.5.5 update-source Loopback0
  neighbor 10.6.6.6 remote-as 64500
  neighbor 10.6.6.6 update-source Loopback0
  no auto-summary
  address-family vpnv4
    bgp advertise-best-external
      neighbor 10.5.5.5 activate
      neighbor 10.5.5.5 send-community extended
      neighbor 10.6.6.6 activate
      neighbor 10.6.6.6 send-community extended
      exit-address-family
    !
    address-family ipv4 vrf test1
      bgp advertise-best-external
      neighbor 10.5.5.5 activate
      neighbor 10.5.5.5 send-community extended
      neighbor 10.6.6.6 activate
      neighbor 10.6.6.6 send-community extended
      exit-address-family
      !
    address-family ipv4 vrf test1
      bgp advertise-best-external
      neighbor 10.5.5.5 activate
      neighbor 10.5.5.5 send-community extended
      neighbor 10.6.6.6 activate
      neighbor 10.6.6.6 send-community extended
      exit-address-family
exit-address-family
```

Example Configuring a Best External Path on an RR for an Intercluster

The following example configures RR1 in Figure 2. RR1 is configured to calculate, install, and advertise the best external path to its intercluster RR neighbors.

```plaintext
router bgp 1
  neighbor 10.5.1.1 remote-as 1
  neighbor 10.5.1.2 remote-as 1
```
address-family ipv4 unicast
neighbor 10.5.1.1 activate
neighbor 10.5.1.2 activate
bgp additional-paths select best-external
bgp additional-paths install
neighbor 10.5.1.1 advertise best-external
neighbor 10.5.1.2 advertise best-external
end

### Additional References

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>Basic MPLS VPNs</td>
<td>“Configuring MPLS Layer 3 VPNs”</td>
</tr>
<tr>
<td>Multiprotocol VRFs</td>
<td>“MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs”</td>
</tr>
<tr>
<td>A failover feature that creates a new path after a link or node failure</td>
<td>“MPLS VPN--BGP Local Convergence”</td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-marques-idr-best-external</td>
<td>BGP Best External, Advertisement of the best external route to iBGP</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2547</td>
<td>BGP/MPLS VPNs</td>
</tr>
</tbody>
</table>
feature information for bgp best external

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>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Best External</td>
<td>Cisco IOS XE Release 2.5</td>
<td>The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. This feature advertises the most preferred route among those received from external neighbors as a backup route. In Cisco IOS XE Release 2.5, this feature was introduced. The following commands were introduced or modified: <code>bgp advertise-best-external</code>, <code>bgp recursion host</code>, <code>show ip bgp</code>, <code>show ip bgp vpnv4</code>, <code>show ip cef</code>, <code>show ip cef vrf</code>, <code>show ip route</code>, <code>show ip route vrf</code></td>
</tr>
<tr>
<td>BGP Best External Path on an RR for Intercluster</td>
<td>Cisco IOS XE Release 3.4S</td>
<td>The BGP Best External Path on RR for Intercluster feature provides path diversity between RR clusters. The feature provides best external functionality toward non-client iBGP peers, and is also known as &quot;intercluster best external path.&quot; The following commands were introduced: <code>bgp additional-pathsselect</code>, <code>neighbor advertise best-external</code></td>
</tr>
</tbody>
</table>
BGP PIC Edge for IP and MPLS-VPN

The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover.

Note

In this document, the BGP PIC Edge for IP and MPLS-VPN feature is called BGP PIC.

- Finding Feature Information, page 559
- Prerequisites for BGP PIC, page 559
- Restrictions for BGP PIC, page 560
- Information About BGP PIC, page 560
- How to Configure BGP PIC, page 568
- Configuration Examples for BGP PIC, page 571
- Additional References, page 574
- Feature Information for BGP PIC, page 575

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Prerequisites for BGP PIC

- Ensure that the Border Gateway Protocol (BGP) and the IP or Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- Ensure that the backup/alternate path has a unique next hop that is not the same as the next hop of the best path.
• Enable the Bidirectional Forwarding Detection (BFD) protocol to quickly detect link failures of directly connected neighbors.

Restrictions for BGP PIC

The following restrictions apply to the BGP PIC feature:

• With BGP Multipath, the BGP Prefix-Independent Convergence (PIC) feature is already supported.
• In MPLS VPNs, the BGP PIC feature is not supported with MPLS VPN Inter-Autonomous Systems Option B.
• The BGP PIC feature supports prefixes only for IPv4, IPv6, VPNv4, and VPNv6 address families.
• The BGP PIC feature cannot be configured with Multicast or L2VPN Virtual Routing and Forwarding (VRF) address families.
• If the route reflector is only in the control plane, then you do not need BGP PIC, because BGP PIC addresses data plane convergence.
• When two PE routers become each other’s backup/alternate path to a CE router, traffic might loop if the CE router fails. Neither router will reach the CE router, and traffic will continue to be forwarded between the PE routers until the time-to-live (TTL) timer expires.
• The BGP PIC feature does not support Nonstop Forwarding with Stateful Switchover (NSF/SSO). However, ISSU is supported if both Route Processors have the BGP PIC feature configured.
• The BGP PIC feature solves the traffic forwarding only for a single network failure at both the edge and the core.
• The BGP PIC feature does not work with the BGP Best External feature. If you try to configure the BGP PIC feature after configuring the BGP Best External feature, you receive an error.

Information About BGP PIC

• Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature, page 560
• How BGP Converges Under Normal Circumstances, page 561
• How BGP PIC Improves Convergence, page 561
• How a Failure Is Detected, page 563
• How BGP PIC Can Achieve Subsecond Convergence, page 563
• How BGP PIC Improves Upon the Functionality of MPLS VPN BGP Local Convergence, page 563
• Configuration Modes for Enabling BGP PIC, page 563
• BGP PIC Scenarios, page 564
• Cisco Express Forwarding Recursion, page 567

Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature

• An additional path for failover allows faster restoration of connectivity if a primary path is invalid or withdrawn.
• Reduction of traffic loss.
• Constant convergence time so that the switching time is the same for all prefixes.
How BGP Converges Under Normal Circumstances

Under normal circumstances, BGP can take several seconds to a few minutes to converge after a network change. At a high level, BGP goes through the following process:

1. BGP learns of failures through either Interior Gateway Protocol (IGP) or BFD events or interface events.
2. BGP withdraws the routes from the routing information base (RIB), and the RIB withdraws the routes from the forwarding information base (FIB) and distributed FIB (dFIB). This process clears the data path for the affected prefixes.
3. BGP sends withdraw messages to its neighbors.
4. BGP calculates the next best path to the affected prefixes.
5. BGP inserts the next best path for affected prefixes into the RIB, and the RIB installs them in the FIB and dFIB.

This process takes a few seconds or a few minutes to complete, depending on the latency of the network, the convergence time across the network, and the local load on the devices. The data plane converges only after the control plane converges.

How BGP PIC Improves Convergence

The BGP PIC functionality is achieved by an additional functionality in the BGP, RIB, Cisco Express Forwarding, and MPLS.

- **BGP Functionality**
  
  BGP PIC affects prefixes under IPv4 and VPNv4 address families. For those prefixes, BGP calculates an additional second best path, along with the primary best path. (The second best path is called the backup/alternate path.) BGP installs the best and backup/alternate paths for the affected prefixes into the BGP RIB. The backup/alternate path provides a fast reroute mechanism to counter a singular network failure. BGP also includes the alternate/backup path in its application programming interface (API) to the IP RIB.

- **RIB Functionality**
  
  For BGP PIC, RIB installs an alternate path per route if one is available. With the BGP PIC functionality, if the RIB selects a BGP route containing a backup/alternate path, it installs the backup/alternate path with the best path. The RIB also includes the alternate path in its API with the FIB.

- **Cisco Express Forwarding Functionality**
  
  With BGP PIC, Cisco Express Forwarding stores an alternate path per prefix. When the primary path goes down, Cisco Express Forwarding searches for the backup/alternate path in a prefix independent manner. Cisco Express Forwarding also listens to BFD events to rapidly detect local failures.

- **MPLS Functionality**
  
  MPLS Forwarding is similar to Cisco Express Forwarding, in that it stores alternate paths and switches to the alternate path if the primary path goes down.

  When the BGP PIC feature is enabled, BGP calculates a backup/alternate path per prefix and installs it into BGP RIB, IP RIB, and FIB. This improves convergence after a network failure. There are two types of network failures that the BGP PIC feature detects:
  
  - Core node/link failure (internal Border Gateway Protocol [iBGP] node failure): If a PE node/link fails, then the failure is detected through IGP convergence. IGP conveys the failure through the RIB to the FIB.
- Local link/immediate neighbor node failure (external Border Gateway Protocol [eBGP] node/link failure): To detect a local link failure or eBGP single-hop peer node failure in less than a second, you must enable BFD. Cisco Express Forwarding looks for BFD events to detect a failure of an eBGP single-hop peer.

Convergence in the Data Plane
Upon detection of a failure, Cisco Express Forwarding detects the alternate next hop for all prefixes affected by the failure. The data plane convergence is achieved in subseconds depending on whether the BGP PIC implementation exists in the software or hardware.

Convergence in the Control Plane
Upon detection of failure, BGP learns about the failure through IGP convergence or BFD events and sends withdraw messages for the prefixes, recalculating the best and backup/alternate paths, and advertising the next best path across the network.

- BGP Fast Reroute Role in the BGP PIC Feature, page 562

BGP Fast Reroute Role in the BGP PIC Feature

BGP Fast Reroute (FRR) provides a best path and a backup/alternate path in BGP, RIB, and Cisco Express Forwarding. BGP FRR provides a very fast reroute mechanism into the RIB and Cisco Express Forwarding on the backup BGP next hop to reach a destination when the current best path is not available. BGP FRR precomputes a second best path in BGP and gives it to the RIB and Cisco Express Forwarding as a backup/alternate path, and Cisco Express Forwarding programs it into line cards. Therefore, BGP FRR sets up the best path and backup/alternate path. The BGP PIC feature provides the ability for Cisco Express Forwarding to quickly switch the traffic to the other egress ports if the current next hop or the link to this next hop goes down. This is illustrated in the figure below.

Figure 49 BGP PIC Edge and BGP FRR

BGP FRR
Selects and stores best path and backup/alternate path per prefix.

BGP RIB

IP RIB

PIC Software
Reacts to next hop failure and switches to alternate path. Convergence not dependent on number of prefixes.

FIB

PIC Hardware
CEF/FIB H/W API

DFIB

DFIB
How a Failure Is Detected

A failure in the iBGP (remote) peer is detected by IGP; it may take a few seconds to detect the failure. Convergence can occur in subseconds or seconds, depending on whether PIC is enabled on the line cards.

If the failure is with directly connected neighbors (eBGP), and if you use BFD to detect when a neighbor has gone down, the detection happens within a subsecond and the convergence can occur in subseconds or seconds, depending on whether PIC is enabled on the line cards.

How BGP PIC Can Achieve Subsecond Convergence

The BGP PIC feature works at the Cisco Express Forwarding level, and Cisco Express Forwarding can be processed in both hardware line cards and in the software.

• For platforms that support Cisco Express Forwarding processing in the line cards, the BGP PIC feature can converge in subseconds. The Cisco 7600 router and Cisco 10000 router supports Cisco Express Forwarding processing in the line cards and in the software, and thus can attain subsecond convergence.
• For platforms that do not use Cisco Express Forwarding in hardware line cards, Cisco Express Forwarding is achieved in the software. The BGP PIC feature will work with the Cisco Express Forwarding through the software and achieve convergence within seconds. The Cisco 7200 router supports Cisco Express Forwarding in the software and thus can achieve convergence in seconds rather than milliseconds.

How BGP PIC Improves Upon the Functionality of MPLS VPN BGP Local Convergence

The BGP PIC feature is an enhancement to the "MPLS VPN--BGP Local Convergence" feature, which provides a failover mechanism that recalculates the best path and installs the new path in forwarding after a link failure. The feature maintains the local label for 5 minutes to ensure that the traffic uses the backup/alternate path, thus minimizing traffic loss.

The BGP PIC feature improves the LoC time to under a second by calculating a backup/alternate path in advance. When a link failure occurs, the traffic is sent to the backup/alternate path.

When you configure the BGP PIC feature, it will override the functionality of the "MPLS VPN--BGP Local Convergence" feature. You do not have to remove the protection local-prefixes command from the configuration.

Configuration Modes for Enabling BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

• VPNv4 address family configuration mode protects all the VRFs.
• VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
• Router configuration mode protects prefixes in the global routing table.
BGP PIC Scenarios

The following scenarios explain how you can configure the BGP PIC functionality to achieve fast convergence:

- IP PE-CE Link and Node Protection on the CE Side (Dual PEs), page 564
- IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes), page 565
- IP MPLS PE-CE Link Protection for the Primary or Backup Alternate Path, page 566
- IP MPLS PE-CE Node Protection for Primary or Backup Alternate Path, page 566

IP PE-CE Link and Node Protection on the CE Side (Dual PEs)

The figure below shows a network that uses the BGP PIC feature. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.

CE1 is configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both routes into the RIB and Cisco Express Forwarding plane. When the CE1-PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate path. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

Figure 50 Using BGP PIC to Protect the PE-CE Link
IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes)

The figure below shows a network that uses the BGP PIC feature on CE1. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.
- An iBGP session exists between the CE1 and CE2 routers.

In this example, CE1 and CE2 are configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both the routes into the RIB and Cisco Express Forwarding plane.

There should not be any policies set on CE1 and CE2 for the eBGP peers PE1 and PE2. Both CE routers must point to the eBGP route as next hop. On CE1, the next hop to reach CE3 is through PE1, so PE1 is the best path to reach CE3. On CE2, the best path to reach CE3 is PE2. CE2 advertises itself as the next hop to CE1, and CE1 does the same to CE2. As a result, CE1 has two paths for the specific prefix and it usually selects the directly connected eBGP path over the iBGP path according to the best path selection rules. Similarly, CE2 has two paths—an eBGP path through PE2 and an iBGP path through CE1-PE1.

When the CE1-PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate node CE2. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

If the CE1-PE1 link or PE1 goes down and BGP PIC is enabled on CE1, BGP recomputes the best path, removing the next hop PE1 from RIB and reinstalling CE2 as the next hop into the RIB and Cisco Express Forwarding. CE1 automatically gets a backup/alternate repair path into Cisco Express Forwarding and the traffic loss during forwarding is now in subseconds, thereby achieving fast convergence.

Figure 51 Using BGP PIC in a Dual CE, Dual PE Network
IP MPLS PE-CE Link Protection for the Primary or Backup Alternate Path

The figure above shows a network that uses the BGP PIC feature on CE1 and CE2. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE routers can be configured with the BGP PIC feature under IPv4 or VPNv4 address families.

For BGP PIC to work in BGP for PE-CE link protection, set the policies on PE3 and PE4 for prefixes received from CE3 so that one of the PE routers acts as the primary and the other as the backup/alternate. Usually, this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. Thus, PE1 has PE3 as the best path and PE4 as the second path.

When the PE3-CE3 link goes down, Cisco Express Forwarding detects the link failure, and PE3 recomputes the best path, selects PE4 as the best path, and sends a withdraw message for the PE3 prefix to the reflect routers. Some of the traffic goes through PE3-PE4 until BGP installs PE4 as the best path route into the RIB and Cisco Express Forwarding. PE1 receives the withdraw, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane.

Thus, with BGP PIC enabled on PE3 and PE4, Cisco Express Forwarding detects the link failure and does in-place modification of the forwarding object to the backup/alternate node PE4 that already exists in Cisco Express Forwarding. PE4 knows that the backup/alternate path is locally generated and routes the traffic to the egress port connected to CE3. This way, traffic loss is minimized and fast convergence is achieved.

IP MPLS PE-CE Node Protection for Primary or Backup Alternate Path

The figure below shows a network that uses the BGP PIC feature on all the PE routers in an MPLS network.

Figure 52  Enabling BGP PIC on All PEs Routers in the MPLS Network
The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE routers are configured with the BGP PIC feature under IPv4 and VPNv4 address families.

For BGP PIC to work in BGP for the PE-CE node protection, set the policies on PE3 and PE4 for the prefixes received from CE3 such that one of the PE routers acts as primary and the other as backup/alternate. Usually, this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. So, PE1 has PE3 as the best path and PE4 as the second path.

When PE3 goes down, PE1 knows about the removal of the host prefix by IGP in subseconds, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane. Normal BGP convergence will happen while BGP PIC is redirecting the traffic through PE4, and packets are not lost.

Thus, with BGP PIC enabled on PE3, Cisco Express Forwarding detects the node failure on PE3 and points the forwarding object to the backup/alternate node PE4. PE4 knows that the backup/alternate path is locally generated and routes the traffic to the egress port using the backup/alternate path. This way, traffic loss is minimized.

No Local Policies Set on the PE Routers

PE1 and PE2 point to the eBGP CE paths as the next hop with no local policy. Each of the PE routers receives the other’s path, and BGP calculates the backup/alternate path and installs it into Cisco Express Forwarding, along with its own eBGP path towards CE as the best path. The limitation of the MPLS PE-CE link and node protection solutions is that you cannot change BGP policies. They should work without the need for a best-external path.

Local Policies Set on the PE Routers

Whenever there is a local policy on the PE routers to select one of the PE routers as the primary path to reach the egress CE, the `bgp advertise-best-external` command is needed on the backup/alternate node PE3 to propagate the external CE routes with a backup/alternate label into the route reflectors and the far-end PE routers.

Cisco Express Forwarding Recursion

Recursion is the ability to find the next longest matching path when the primary path goes down.

When the BGP PIC feature is not installed, and if the next hop to a prefix fails, Cisco Express Forwarding finds the next path to reach the prefix by recursing through the FIB to find the next longest matching path to the prefix. This is useful if the next hop is multiple hops away and there is more than one way of reaching the next hop.

However, with the BGP PIC feature, you may want to disable Cisco Express Forwarding recursion for the following reasons:
• Recursion slows down convergence when Cisco Express Forwarding searches all the FIB entries.
• BGP PIC Edge already precomputes an alternate path, thus eliminating the need for Cisco Express Forwarding recursion.

When the BGP PIC functionality is enabled, Cisco Express Forwarding recursion is disabled by default for two conditions:
• For next hops learned with a /32 network mask (host routes)
• For next hops that are directly connected

For all other cases, Cisco Express Forwarding recursion is enabled.

As part of the BGP PIC functionality, you can issue the `bgp recursion host` command to disable or enable Cisco Express Forwarding recursion for BGP host routes.

To disable or enable Cisco Express Forwarding recursion for BGP directly connected next hops, you can issue the `disable-connected-check` command.

### How to Configure BGP PIC

- Configuring BGP PIC, page 568

### Configuring BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

For a full configuration example that includes configuring multiprotocol VRFs and shows output to verify that the feature is enabled, see the Example Configuring BGP PIC, page 571.

- If you are implementing the BGP PIC feature in an MPLS VPN, ensure that the network is working properly before configuring the BGP PIC feature. See "Configuring MPLS Layer 3 VPNs" for more information.
- If you are implementing the BGP PIC feature in an MPLS VPN, configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs".
- Ensure that the CE router is connected to the network by at least two paths.
SUMMARY STEPS

1. enable
2. configure terminal
3. **router bgp** autonomous-system-number
4. Do one of the following:
   • address-family ipv4 [unicast | vrf vrf-name]
   • or
   • address-family vpnv4 [unicast]
5. bgp additional-paths install
6. neighbor *ip-address* remote-as autonomous-system-number
7. neighbor *ip-address* activate
8. bgp recursion host
9. neighbor *ip-address* fall-over [bfd | route-map map-name]
10. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# router bgp 40000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 4** Do one of the following:  
- `address-family ipv4 [unicast | vrf vrf-name]`  
- or  
- `address-family vpv4 [unicast]`  

**Example:**  
Router(config-router)# address-family ipv4 unicast

- The `unicast` keyword specifies the IPv4 or VVPNv4 unicast address family.  
- The `vrf` keyword and `vrf-name` argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.

**Step 5** `bgp additional-paths install`  

**Example:**  
Router(config-router-af)# bgp additional-paths install

Calculates a backup/alternate path and installs it into the RIB and Cisco Express Forwarding.

**Step 6** `neighbor ip-address remote-as autonomous-system-number`  

**Example:**  
Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000

Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.  
- By default, neighbors that are defined using the `neighbor remote-as` command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the `neighbor activate` command in address family configuration mode for the other prefix types.

**Step 7** `neighbor ip-address activate`  

**Example:**  
Router(config-router-af)# neighbor 192.168.1.1 activate

Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
### Command or Action | Purpose
--- | ---
Step 8 | bgp recursion host
**Example:**
```
Router(config-router-af)# bgp recursion host
```
(Optional) Enables the recursive-via-host flag for IPv4, VPNv4, and VRF address families.
- When the BGP PIC feature is enabled, Cisco Express Forwarding recursion is disabled. Under most circumstances, you do not want to enable recursion when BGP PIC is enabled.

Step 9 | neighbor ip-address fall-over [bfd route-map map-name]
**Example:**
```
Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd
```
Enables BFD protocol support to detect when a neighbor has gone away, which can occur within a subsecond.

Step 10 | end
**Example:**
```
Router(config-router-af)# end
```
Exits address family configuration mode and returns to privileged EXEC mode.

---

### Configuration Examples for BGP PIC

- Example Configuring BGP PIC, page 571
- Example Displaying Backup Alternate Paths for BGP PIC, page 572

### Example Configuring BGP PIC

The following example shows how to configure the BGP PIC feature in VPNv4 address family configuration mode, which enables the feature on all VRFs. In the following example, there are two VRFs defined: blue and green. All the VRFs, including those in VRFs blue and green, are protected by backup/alternate paths.

```plaintext
vrf definition test1
  rd 400:1
  route-target export 100:1
  route-target export 200:1
  route-target export 300:1
  route-target export 400:1
  route-target import 100:1
  route-target import 200:1
  route-target import 300:1
  route-target import 400:1
  address-family ipv4
  exit-address-family
  exit
! interface Ethernet1/0
  vrf forwarding test1
  ip address 10.0.0.1 255.0.0.0
```
exit
router bgp 3
no synchronization
bgp log-neighbor-changes
redistribute static
redistribute connected
neighbor 10.6.6.6 remote-as 3
neighbor 10.6.6.6 update-source Loopback0
neighbor 10.7.7.7 remote-as 3
neighbor 10.7.7.7 update-source Loopback0
no auto-summary
!
address-family vpnv4
bgp additional-paths install
neighbor 10.6.6.6 activate
neighbor 10.6.6.6 send-community both
neighbor 10.7.7.7 activate
neighbor 10.7.7.7 send-community both
exit-address-family
!
address-family ipv4 vrf blue
import path selection all
import path limit 10
no synchronization
neighbor 10.11.11.11 remote-as 1
neighbor 10.11.11.11 activate
exit-address-family
!
address-family ipv4 vrf green
import path selection all
import path limit 10
no synchronization
neighbor 10.13.13.13 remote-as 1
neighbor 10.13.13.13 activate
exit-address-family

The following show vrf detail command output shows that the BGP PIC feature is enabled:

Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
Interfaces:
Se4/0
Address family ipv4 (Table ID = 1 (0x1)):
  Export VPN route-target communities
    RT:100:1    RT:200:1    RT:300:1
    RT:400:1
  Import VPN route-target communities
    RT:100:1    RT:200:1    RT:300:1
    RT:400:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
  Prefix protection with additional path enabled
  Address family ipv6 not active.

Example Displaying Backup Alternate Paths for BGP PIC

The command output in the following example shows that the VRFs in VRF blue have backup/alternate paths:

Router# show ip bgp vpnv4 vrf blue 10.0.0.0
BGP routing table entry for 10:12:12.0.0.0/24, version 88
Paths: (4 available, best #1, table blue)
  Additional-path
  Advertised to update-groups: 6
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
The command output in the following example shows that the VRFs in VRF green have backup/alternate paths:

```
Router# show ip bgp vpnv4 vrf green 12.0.0.0
BGP routing table entry for 12:23:12.0.0.0/24, version 87
Paths: (4 available, best #4, table green)
   Additional-path
   Advertised to update-groups:
   5
   1, imported path from 12:23:12.0.0.0/24
   10.13.13.13 (via green) from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:12:23 , recursive-via-connected
1, imported path from 12:23:12.0.0.0/24
   10.3.3.3 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 200, valid, internal
      Extended Community: RT:12:23
   Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
   mpls labels in/out nolabel/37
1
10.11.11.11 from 10.11.11.11 (1.0.0.1)
   Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
   Extended Community: RT:12:23 , recursive-via-connected
```

The command output in the following example shows the BGP routing table entries for the backup and alternate paths:

```
Router# show ip bgp 10.0.0.0 255.255.0.0
BGP routing table entry for 10.0.0.0/16, version 123
Paths: (4 available, best #3, table default)
   Additional-path
   Advertised to update-groups:
   2
   Local
10.0.101.4 from 10.0.101.4 (10.3.3.3)
   Origin IGP, localpref 100, weight 500, valid, internal
Local
10.0.101.3 from 10.0.101.3 (10.4.4.4)
   Origin IGP, localpref 100, weight 200, valid, internal
Local
10.0.101.2 from 10.0.101.2 (10.1.1.1)
   Origin IGP, localpref 100, weight 900, valid, internal, best
Local
10.0.101.1 from 10.0.101.1 (10.5.5.5)
   Origin IGP, localpref 100, weight 700, valid, internal, backup/repair
```
The command output in the following example shows the routing information base entries for the backup and alternate paths:

```
Router# show ip route repair-paths 10.0.0.0 255.255.0.0
Routing entry for 10.0.0.0/16
    Known via "bgp 10", distance 200, metric 0, type internal
    Last update from 10.0.101.2 00:00:56 ago
    Routing Descriptor Blocks:
        * 10.0.101.2, from 10.0.101.2, 00:00:56 ago
            Route metric is 0, traffic share count is 1
            AS Hops 0
            MPLS label: none
        [RPR]10.0.101.1, from 10.0.101.1, 00:00:56 ago
            Route metric is 0, traffic share count is 1
            AS Hops 0
            MPLS label: none
```

The command output in the following example shows the Cisco Express Forwarding/forwarding information base entries for the backup and alternate paths:

```
Router# show ip cef 10.0.0.0 255.255.0.0 detail
10.0.0.0/16, epoch 0, flags rib only nolabel, rib defined all labels
    recursive via 10.0.101.2
        attached to GigabitEthernet0/2
    recursive via 10.0.101.1, repair
        attached to GigabitEthernet0/2
```

### Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>BGP commands: complete command syntax,</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
<tr>
<td>command mode, command history, defaults, usage</td>
<td></td>
</tr>
<tr>
<td>guidelines, and examples</td>
<td></td>
</tr>
<tr>
<td>Basic MPLS VPNs</td>
<td>&quot;Configuring MPLS Layer 3 VPNs&quot;</td>
</tr>
<tr>
<td>A failover feature that creates a new path after</td>
<td>&quot;MPLS VPN--BGP Local Convergence&quot;</td>
</tr>
<tr>
<td>a link or node failure</td>
<td></td>
</tr>
<tr>
<td>Configuring multiprotocol VRFs</td>
<td>&quot;MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs&quot;</td>
</tr>
</tbody>
</table>

### Related Documents

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td>draft-walton-bgp-add-paths-04.txt</td>
<td>Advertisement of Multiple Paths in BGP</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2547</td>
<td>BGP/MPLS VPNs</td>
</tr>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP-4)</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

Feature Information for BGP PIC

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 37  Feature Information for BGP PIC

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP PIC Edge for IP and MPLS-VPN</td>
<td>Cisco IOS XE Release 2.5</td>
<td>The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover. In Cisco IOS XE Release 2.5, this feature was introduced. The following commands were introduced or modified: bgp additional-paths install, bgp recursion host, show ip bgp, show ip cef, show ip route, show vrf.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.