



IP Routing: BGP Configuration Guide, Cisco IOS Release 15SY

First Published: October 15, 2012

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

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 ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT 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.

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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <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. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Cisco BGP Overview 1

- Finding Feature Information 1
- Prerequisites for Cisco BGP 1
- Restrictions for Cisco BGP 2
- Information About Cisco BGP 2
 - BGP Version 4 2
 - BGP Version 4 Functional Overview 2
 - BGP Autonomous Systems 3
 - BGP Autonomous System Number Formats 4
 - Classless Interdomain Routing 6
 - Multiprotocol BGP 7
 - Benefits of Using Multiprotocol BGP Versus BGP 7
 - Multiprotocol BGP Extensions for IP Multicast 7
 - NLRI Configuration CLI 9
 - Cisco BGP Address Family Model 10
 - IPv4 Address Family 12
 - IPv6 Address Family 12
 - CLNS Address Family 13
 - VPNv4 Address Family 13
 - L2VPN Address Family 14
 - BGP CLI Removal Considerations 15
- Additional References 16
- Feature Information for Cisco BGP Overview 17

CHAPTER 2

BGP 4 19

- Finding Feature Information 19
- Information About BGP 4 19
 - BGP Version 4 Functional Overview 19

BGP Router ID	20
BGP-Speaker and Peer Relationships	21
BGP Peer Session Establishment	21
BGP Session Reset	22
BGP Route Aggregation	22
BGP Route Aggregation Generating AS_SET Information	23
Routing Policy Change Management	23
BGP Peer Groups	24
BGP Backdoor Routes	24
How to Configure BGP 4	25
Configuring a BGP Routing Process	25
Troubleshooting Tips	28
Configuring a BGP Peer	28
Troubleshooting Tips	32
Configuring a BGP Peer for the IPv4 VRF Address Family	32
Troubleshooting Tips	36
Customizing a BGP Peer	36
Removing BGP Configuration Commands Using a Redistribution	41
Monitoring and Maintaining Basic BGP	43
Configuring Inbound Soft Reconfiguration When Route Refresh Capability Is Missing	43
Resetting and Displaying Basic BGP Information	46
Aggregating Route Prefixes Using BGP	47
Redistributing a Static Aggregate Route into BGP	48
Configuring Conditional Aggregate Routes Using BGP	49
Suppressing and Unsuppressing the Advertisement of Aggregated Routes Using BGP	50
Conditionally Advertising BGP Routes	52
Originating BGP Routes	55
Advertising a Default Route Using BGP	56
Originating BGP Routes Using Backdoor Routes	57
Configuring a BGP Peer Group	59
Configuration Examples for BGP 4	61
Example: Configuring a BGP Process and Customizing Peers	61
Examples: Removing BGP Configuration Commands Using a Redistribution Example	62

Examples: BGP Soft Reset	63
Example: Resetting and Displaying Basic BGP Information	63
Examples: Aggregating Prefixes Using BGP	64
Example: Configuring a BGP Peer Group	65
Additional References	66
Feature Information for BGP 4	67

CHAPTER 3

Configuring a Basic BGP Network	69
Finding Feature Information	69
Prerequisites for Configuring a Basic BGP Network	70
Restrictions for Configuring a Basic BGP Network	70
Information About Configuring a Basic BGP Network	70
BGP Version 4	70
BGP Router ID	70
BGP-Speaker and Peer Relationships	71
BGP Autonomous System Number Formats	71
Cisco Implementation of 4-Byte Autonomous System Numbers	73
BGP Peer Session Establishment	74
Cisco Implementation of BGP Global and Address Family Configuration Commands	75
BGP Session Reset	76
BGP Route Aggregation	77
BGP Aggregation Route AS_SET Information Generation	77
Routing Policy Change Management	78
Conditional BGP Route Injection	79
BGP Peer Groups	79
BGP Backdoor Routes	80
Peer Groups and BGP Update Messages	80
BGP Update Group	81
BGP Dynamic Update Group Configuration	81
BGP Peer Templates	81
Inheritance in Peer Templates	82
Peer Session Templates	82
Peer Policy Templates	84
BGP IPv6 Neighbor Activation Under the IPv4 Address Family	85
How to Configure a Basic BGP Network	85

Configuring a BGP Routing Process	85
Troubleshooting Tips	88
Configuring a BGP Peer	88
Troubleshooting Tips	92
What to Do Next	92
Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers	92
Troubleshooting Tips	95
Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers	95
Configuring a BGP Peer for the IPv4 VRF Address Family	99
Troubleshooting Tips	102
Customizing a BGP Peer	102
Removing BGP Configuration Commands Using a Redistribution	108
Monitoring and Maintaining Basic BGP	109
Configuring Inbound Soft Reconfiguration When Route Refresh Capability Is Missing	110
Resetting and Displaying Basic BGP Information	113
Aggregating Route Prefixes Using BGP	114
Redistributing a Static Aggregate Route into BGP	114
Configuring Conditional Aggregate Routes Using BGP	116
Suppressing and Unsuppressing the Advertisement of Aggregated Routes Using BGP	117
Suppressing Inactive Route Advertisement Using BGP	119
Conditionally Advertising BGP Routes	121
Originating BGP Routes	124
Advertising a Default Route Using BGP	124
Troubleshooting Tips	126
Conditionally Injecting BGP Routes	126
Troubleshooting Tips	130
Originating BGP Routes Using Backdoor Routes	130
Configuring a BGP Peer Group	131
Configuring Peer Session Templates	134
Configuring a Basic Peer Session Template	134
What to Do Next	136

Configuring Peer Session Template Inheritance with the inherit peer-session	
Command	136
What to Do Next	138
Configuring Peer Session Template Inheritance with the neighbor inherit peer-session	
Command	138
What to Do Next	140
Configuring Peer Policy Templates	140
Configuring Basic Peer Policy Templates	140
What to Do Next	142
Configuring Peer Policy Template Inheritance with the inherit peer-policy Command	142
Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy	
Command	144
Monitoring and Maintaining BGP Dynamic Update Groups	147
Troubleshooting Tips	148
Configuration Examples for a Basic BGP Network	148
Example: Configuring a BGP Process and Customizing Peers	148
Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System	
Numbers	148
Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte	
Autonomous System Number	151
Example: NLRI to AFI Configuration	153
Examples: Removing BGP Configuration Commands Using a Redistribution Example	154
Examples: BGP Soft Reset	155
Example: Resetting BGP Peers Using 4-Byte Autonomous System Numbers	156
Example: Resetting and Displaying Basic BGP Information	156
Examples: Aggregating Prefixes Using BGP	158
Example: Configuring a BGP Peer Group	159
Example: Configuring Peer Session Templates	159
Examples: Configuring Peer Policy Templates	159
Examples: Monitoring and Maintaining BGP Dynamic Update Peer-Groups	160
Where to Go Next	161
Additional References	162
Feature Information for Configuring a Basic BGP Network	163

Finding Feature Information	167
Information About BGP Support for 4-byte ASN	167
BGP Autonomous System Number Formats	167
Cisco Implementation of 4-Byte Autonomous System Numbers	170
How to Configure BGP Support for 4-byte ASN	171
Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers	171
Troubleshooting Tips	174
Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers	174
Configuration Examples for BGP Support for 4-byte ASN	177
Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers	177
Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number	180
Additional References for BGP Support for 4-byte ASN	182
Feature Information for BGP Support for 4-byte ASN	183

CHAPTER 5**IPv6 Routing: Multiprotocol BGP Extensions for IPv6 185**

Finding Feature Information	185
Information About IPv6 Routing: Multiprotocol BGP Extensions for IPv6	185
Multiprotocol BGP Extensions for IPv6	185
How to Implement Multiprotocol BGP for IPv6	186
Configuring an IPv6 BGP Routing Process and BGP Router ID	186
Configuring IPv6 Multiprotocol BGP Between Two Peers	187
Advertising Routes into IPv6 Multiprotocol BGP	188
Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes	190
Redistributing Prefixes into IPv6 Multiprotocol BGP	192
Clearing External BGP Peers	194
Advertising IPv4 Routes Between IPv6 BGP Peers	194
Configuration Examples for Multiprotocol BGP for IPv6	197
Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer	197
Example: Configuring an IPv6 Multiprotocol BGP Peer Group	197
Example: Advertising Routes into IPv6 Multiprotocol BGP	197

Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes	197
Example: Redistributing Prefixes into IPv6 Multiprotocol BGP	198
Example: Advertising IPv4 Routes Between IPv6 Peers	198
Additional References	198
Feature Information for IPv6 Routing: Multiprotocol BGP Extensions for IPv6	199

CHAPTER 6**IPv6 Routing: Multiprotocol BGP Link-Local Address Peering 201**

Finding Feature Information	201
Information About IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	201
IPv6 Multiprotocol BGP Peering Using a Link-Local Address	201
How to Configure IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	202
Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address	202
Configuration Examples for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	206
Example: Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address	206
Additional References	207
Feature Information for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	208

CHAPTER 7**IPv6 Multicast Address Family Support for Multiprotocol BGP 209**

Finding Feature Information	209
Information About IPv6 Multicast Address Family Support for Multiprotocol BGP	209
Multiprotocol BGP for the IPv6 Multicast Address Family	209
How to Implement IPv6 Multicast Address Family Support for Multiprotocol BGP	210
Configuring an IPv6 Peer Group to Perform Multicast BGP Routing	210
Advertising Routes into IPv6 Multiprotocol BGP	212
Redistributing Prefixes into IPv6 Multiprotocol BGP	213
Assigning a BGP Administrative Distance	214
Generating Translate Updates for IPv6 Multicast BGP	215
Resetting IPv6 BGP Sessions	217
Clearing External BGP Peers	217
Clearing IPv6 BGP Route Dampening Information	218
Clearing IPv6 BGP Flap Statistics	219
Configuration Examples for IPv6 Multicast Address Family Support for Multiprotocol BGP	219
Example: Configuring an IPv6 Multiprotocol BGP Peer Group	219
Example: Advertising Routes into IPv6 Multiprotocol BGP	220
Example: Redistributing Prefixes into IPv6 Multiprotocol BGP	220

Example: Generating Translate Updates for IPv6 Multicast BGP	220
Additional References	220
Feature Information for IPv6 Multicast Address Family Support for Multiprotocol BGP	221

CHAPTER 8**Connecting to a Service Provider Using External BGP 223**

Finding Feature Information	223
Prerequisites for Connecting to a Service Provider Using External BGP	224
Restrictions for Connecting to a Service Provider Using External BGP	224
Information About Connecting to a Service Provider Using External BGP	224
External BGP Peering	224
BGP Autonomous System Number Formats	225
BGP Attributes	228
Multihoming	230
MED Attribute	230
Transit Versus Nontransit Traffic	230
BGP Policy Configuration	231
BGP COMMUNITIES Attribute	232
Extended Communities	232
Extended Community Lists	233
Administrative Distance	233
BGP Route Map Policy Lists	234
How to Connect to a Service Provider Using External BGP	234
Influencing Inbound Path Selection	234
Influencing Inbound Path Selection by Modifying the AS_PATH Attribute	235
Influencing Inbound Path Selection by Setting the MED Attribute	239
Influencing Outbound Path Selection	243
Influencing Outbound Path Selection Using the Local_Pref Attribute	243
Filtering Outbound BGP Route Prefixes	246
Configuring BGP Peering with ISPs	249
Configuring Multihoming with Two ISPs	249
Multihoming with a Single ISP	253
Configuring Multihoming to Receive the Full Internet Routing Table	260
Configuring BGP Policies	264
Filtering BGP Prefixes with Prefix Lists	264
Filtering BGP Prefixes with AS-Path Filters	268

Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers	271
Filtering Traffic Using Community Lists	275
Filtering Traffic Using Extended Community Lists	280
Filtering Traffic Using a BGP Route Map Policy List	283
Filtering Traffic Using Continue Clauses in a BGP Route Map	288
Configuration Examples for Connecting to a Service Provider Using External BGP	291
Example: Influencing Inbound Path Selection	291
Example: Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers	292
Example: Filtering BGP Prefixes with Prefix Lists	293
Example: Filtering BGP Prefixes Using a Single Prefix List	293
Example: Filtering BGP Prefixes Using a Group of Prefixes	294
Example: Adding or Deleting Prefix List Entries	295
Example: Filtering Traffic Using COMMUNITIES Attributes	295
Example: Filtering Traffic Using AS-Path Filters	296
Example: Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers	296
Example: Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers	297
Example: Filtering Traffic Using a BGP Route Map	300
Where to Go Next	300
Additional References	300
Feature Information for Connecting to a Service Provider Using External BGP	302

CHAPTER 9
BGP Named Community Lists 307

Finding Feature Information	307
Information About BGP Named Community Lists	307
BGP COMMUNITIES Attribute	307
BGP Community Lists	308
How to Use BGP Named Community Lists	308
Filtering Traffic Using Community Lists	308
Filtering Traffic Using Extended Community Lists	314
Configuration Examples for BGP Named Community Lists	318
Example: Filtering Traffic Using COMMUNITIES Attributes	318

Additional References for BGP Named Community Lists 318

Feature Information for BGP Named Community Lists 319

CHAPTER 10

BGP Prefix-Based Outbound Route Filtering 321

Finding Feature Information 321

Information About BGP Prefix-Based Outbound Route Filtering 321

 BGP Prefix-Based Outbound Route Filtering 321

How to Configure BGP Prefix-Based Outbound Route Filtering 322

 Filtering Outbound Routes Based on BGP Prefix 322

Configuration Examples for BGP Prefix-Based Outbound Route Filtering 325

 Example: Influencing Outbound Path Selection 325

Additional References 326

Feature Information for BGP Prefix-Based Outbound Route Filtering 327

CHAPTER 11

BGP Route-Map Continue Support for Outbound Policy 329

Finding Feature Information 329

Information About BGP Route-Map Continue Support for Outbound Policy 330

 BGP Route Map with a Continue Clause 330

 Route Map Operation Without Continue Clauses 330

 Route Map Operation with Continue Clauses 330

 Match Operations with Continue Clauses 330

 Set Operations with Continue Clauses 331

How to Filter Traffic Using Continue Clauses in a BGP Route Map 332

 Filtering Traffic Using Continue Clauses in a BGP Route Map 332

Configuration Examples for BGP Route-Map Continue Support for Outbound Policy 335

 Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map 335

Additional References 337

Feature Information for BGP Route-Map Continue Support for Outbound Policy 337

CHAPTER 12

Removing Private AS Numbers from the AS Path in BGP 339

Finding Feature Information 339

Restrictions on Removing and Replacing Private ASNs from the AS Path 339

Information About Removing and Replacing Private ASNs from the AS Path 340

 Public and Private AS Numbers 340

 Benefit of Removing and Replacing Private ASNs from the AS Path 340

Former Restrictions to Removing Private ASNs from the AS Path	340
Enhancements to Removing Private ASNs from the AS Path	340
How to Remove and Replace Private ASNs from the AS Path	341
Removing and Replacing Private ASNs from the AS Path (Cisco IOS Release 15.1(2)T and Later)	341
Configuration Examples for Removing and Replacing Private ASNs from the AS Path	345
Example: Removing Private ASNs (Cisco IOS Release 15.1(2)T)	345
Example: Removing and Replacing Private ASNs (Cisco IOS Release 15.1(2)T)	346
Example: Removing Private ASNs (Cisco IOS Release 12.2)	347
Additional References	348
Feature Information for Removing and Replacing Private ASNs from the AS Path	349

CHAPTER 13

Configuring BGP Neighbor Session Options	351
Finding Feature Information	351
Information About Configuring BGP Neighbor Session Options	352
BGP Neighbor Sessions	352
BGP Support for Fast Peering Session Deactivation	352
BGP Hold Timer	352
BGP Fast Peering Session Deactivation	352
Selective Address Tracking for BGP Fast Session Deactivation	352
BFD Support of BGP IPv6 Neighbors	353
TTL Security Check for BGP Neighbor Sessions	353
BGP Support for the TTL Security Check	353
TTL Security Check for BGP Neighbor Sessions	353
TTL Security Check Support for Multihop BGP Neighbor Sessions	354
Benefits of the BGP Support for TTL Security Check	354
BGP Support for TCP Path MTU Discovery per Session	354
Path MTU Discovery	354
BGP Neighbor Session TCP PMTUD	355
How to Configure BGP Neighbor Session Options	355
Configuring Fast Session Deactivation	355
Configuring Fast Session Deactivation for a BGP Neighbor	355
Configuring Selective Address Tracking for Fast Session Deactivation	357
What to Do Next	360
Configuring BFD for BGP IPv6 Neighbors	360

Configuring the TTL Security Check for BGP Neighbor Sessions	362
Configuring BGP Support for TCP Path MTU Discovery per Session	366
Disabling TCP Path MTU Discovery Globally for All BGP Sessions	366
Disabling TCP Path MTU Discovery for a Single BGP Neighbor	369
Enabling TCP Path MTU Discovery Globally for All BGP Sessions	371
Enabling TCP Path MTU Discovery for a Single BGP Neighbor	373
Configuration Examples for BGP Neighbor Session Options	375
Example: Configuring Fast Session Deactivation for a BGP Neighbor	375
Example: Configuring Selective Address Tracking for Fast Session Deactivation	376
Example: Configuring BFD for a BGP IPv6 Neighbor	376
Example: Configuring the TTL-Security Check	376
Examples: Configuring BGP Support for TCP Path MTU Discovery per Session	377
Example: Disabling TCP Path MTU Discovery Globally for All BGP Sessions	377
Example: Disabling TCP Path MTU Discovery for a Single BGP Neighbor	377
Example: Enabling TCP Path MTU Discovery Globally for All BGP Sessions	377
Example: Enabling TCP Path MTU Discovery for a Single BGP Neighbor	378
Where to Go Next	378
Additional References	378
Feature Information for Configuring BGP Neighbor Session Options	380

CHAPTER 14
BGP Neighbor Policy 383

Finding Feature Information	383
Information About BGP Neighbor Policy	383
Benefit of BGP Neighbor Policy Feature	383
How to Display BGP Neighbor Policy Information	384
Displaying BGP Neighbor Policy Information	384
Additional References	384
Feature Information for BGP Neighbor Policy	385

CHAPTER 15
BGP Dynamic Neighbors 387

Finding Feature Information	387
Information About BGP Dynamic Neighbors	387
BGP Dynamic Neighbors	387
How to Configure BGP Dynamic Neighbors	388
Implementing BGP Dynamic Neighbors Using Subnet Ranges	388

Configuration Examples for BGP Dynamic Neighbors	395
Example: Implementing BGP Dynamic Neighbors Using Subnet Ranges	395
Additional References	396
Feature Information for BGP Dynamic Neighbors	397
<hr/>	
CHAPTER 16	BGP Support for Next-Hop Address Tracking 399
Finding Feature Information	399
Information About BGP Support for Next-Hop Address Tracking	399
BGP Next-Hop Address Tracking	399
Default BGP Scanner Behavior	400
BGP Next_Hop Attribute	400
Selective BGP Next-Hop Route Filtering	400
BGP Support for Fast Peering Session Deactivation	401
BGP Hold Timer	401
BGP Fast Peering Session Deactivation	401
Selective Address Tracking for BGP Fast Session Deactivation	401
How to Configure BGP Support for Next-Hop Address Tracking	401
Configuring BGP Next-Hop Address Tracking	401
Configuring BGP Selective Next-Hop Route Filtering	401
Adjusting the Delay Interval for BGP Next-Hop Address Tracking	405
Disabling BGP Next-Hop Address Tracking	406
Configuring Fast Session Deactivation	407
Configuring Fast Session Deactivation for a BGP Neighbor	408
Configuring Selective Address Tracking for Fast Session Deactivation	409
Configuration Examples for BGP Support for Next-Hop Address Tracking	412
Example: Enabling and Disabling BGP Next-Hop Address Tracking	412
Example: Adjusting the Delay Interval for BGP Next-Hop Address Tracking	412
Examples: Configuring BGP Selective Next-Hop Route Filtering	412
Example: Configuring Fast Session Deactivation for a BGP Neighbor	413
Example: Configuring Selective Address Tracking for Fast Session Deactivation	413
Additional References	413
Feature Information for BGP Support for Next-Hop Address Tracking	414

CHAPTER 17**BGP Restart Neighbor Session After Max-Prefix Limit Reached 417**

Finding Feature Information	417
-----------------------------	-----

Information About BGP Neighbor Session Restart After Max-Prefix Limit Reached	418
Prefix Limits and BGP Peering Sessions	418
BGP Neighbor Session Restart with the Maximum Prefix Limit	418
Subcodes for BGP Cease Notification	418
How to Configure a Device to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded	419
Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded	419
Troubleshooting Tips	422
Configuration Example for BGP Restart Neighbor Session After Max-Prefix Limit Reached	422
Example: Restart Session After Maximum Number of Prefixes from Neighbor Reached	422
Additional References for BGP Restart Neighbor Session After Max-Prefix Limit Reached	422
Feature Information for BGP Restart Neighbor Session after Max-Prefix Limit	423

CHAPTER 18**BGP Support for Dual AS Configuration for Network AS Migrations 425**

Finding Feature Information	425
Information About BGP Support for Dual AS Configuration for Network AS Migrations	426
Autonomous System Migration for BGP Networks	426
Dual Autonomous System Support for BGP Network Autonomous System Migration	426
BGP Network Migration to 4-Byte Autonomous System Numbers	427
How to Configure BGP Support for Dual AS Configuration for Network AS Migrations	427
Configuring Dual AS Peering for Network Migration	427
Configuration Examples for Dual-AS Peering for Network Migration	430
Example: Dual AS Configuration	430
Example: Dual AS Confederation Configuration	431
Example: Replace an AS with Another AS in Routing Updates	431
Additional References	431
Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations	432

CHAPTER 19**Configuring Internal BGP Features 433**

Finding Feature Information	433
Information About Internal BGP Features	433

BGP Routing Domain Confederation	433
BGP Route Reflector	434
Route Reflector Mechanisms to Avoid Routing Loops	437
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer	437
BGP Route Dampening	438
Route Dampening Minimizes Route Flapping	438
BGP Route Dampening Terms	438
BGP Route Map Next Hop Self	439
How to Configure Internal BGP Features	439
Configuring a Routing Domain Confederation	439
Configuring a Route Reflector	440
Configuring a Route Reflector Using a Route Map to a Set Next Hop for an iBGP Peer	441
Adjusting BGP Timers	444
Configuring the Router to Consider a Missing MED as the Worst Path	445
Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths	445
Configuring the Router to Use the MED to Choose a Path in a Confederation	446
Enabling and Configuring BGP Route Dampening	446
Monitoring and Maintaining BGP Route Dampening	448
Configuring BGP Route Map next-hop self	449
Configuration Examples for Internal BGP Features	453
Example: BGP Confederation Configurations with Route Maps	453
Example: BGP Confederation	454
Example: Route Reflector Using a Route Map to Set a Next Hop for an iBGP Peer	455
Example: Configuring BGP Route Map next-hop self	455
Additional References for Internal BGP Features	456
Feature Information for Configuring Internal BGP Features	458

CHAPTER 20

BGP VPLS Auto Discovery Support on Route Reflector	461
Finding Feature Information	461
Information About BGP VPLS Auto Discovery Support on Route Reflector	461
BGP VPLS Autodiscovery Support on Route Reflector	461
Configuration Example for BGP VPLS Auto Discovery Support on Route Reflector	462
Example: BGP VPLS Autodiscovery Support on Route Reflector	462
Additional References	462

Feature Information for BGP VPLS Auto Discovery Support on Route Reflector 463

CHAPTER 21

BGP NSF Awareness 465

Finding Feature Information 465

Information About BGP NSF Awareness 466

 Cisco NSF Routing and Forwarding Operation 466

 Cisco Express Forwarding for NSF 466

 BGP Graceful Restart for NSF 467

 BGP NSF Awareness 467

How to Configure BGP NSF Awareness 468

 Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart 468

 Enabling BGP Global NSF Awareness Using BGP Graceful Restart 468

 Troubleshooting Tips 469

 What to Do Next 470

 Configuring BGP NSF Awareness Timers 470

 What to Do Next 471

 Verifying the Configuration of BGP Nonstop Forwarding Awareness 471

Configuration Examples for BGP NSF Awareness 473

 Example: Enabling BGP Global NSF Awareness Using Graceful Restart 473

Additional References 473

Feature Information for BGP NSF Awareness 474

CHAPTER 22

IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family 477

Finding Feature Information 477

Information About IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family 477

 Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family 477

How to Configure IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family 478

 Configuring the IPv6 BGP Graceful Restart Capability 478

Configuration Examples for IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family 479

 Example: Configuring the IPv6 BGP Graceful Restart Capability 479

Additional References 479

Feature Information for IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family 480

CHAPTER 23**BGP Support for BFD 483**

- Finding Feature Information 483
- Information About BGP Support for BFD 483
 - BFD for BGP 483
- How to Decrease BGP Convergence Time Using BFD 484
 - Prerequisites 484
 - Restrictions 484
 - Decreasing BGP Convergence Time Using BFD 484
 - Configuring BFD Session Parameters on the Interface 485
 - Configuring BFD Support for BGP 486
 - Monitoring and Troubleshooting BFD 487
- Additional References 488
- Feature Information for BGP Support for BFD 489

CHAPTER 24**BGP Support for MTR 491**

- Finding Feature Information 491
- Information About BGP Support for MTR 492
 - BGP Network Scope 492
 - MTR CLI Hierarchy Under BGP 492
 - BGP Sessions for Class-Specific Topologies 493
 - Topology Translation Using BGP 493
 - Topology Import Using BGP 493
- How to Configure BGP Support for MTR 493
 - Configuring BGP Support for MTR 493
 - Activating an MTR Topology Using BGP 494
 - What to Do Next 498
 - Importing Routes from an MTR Topology Using BGP 498
- Configuration Examples for BGP Support for MTR 501
 - Example: Importing Routes from an MTR Topology Using BGP 501
 - Example: Activating an MTR Topology Using BGP 502
- Additional References 504
- Feature Information for BGP Support for MTR 504

CHAPTER 25**BGP Link Bandwidth 507**

Finding Feature Information	507
Prerequisites for BGP Link Bandwidth	508
Restrictions for BGP Link Bandwidth	508
Information About BGP Link Bandwidth	508
BGP Link Bandwidth Overview	508
Link Bandwidth Extended Community Attribute	508
Benefits of the BGP Link Bandwidth Feature	509
How to Configure BGP Link Bandwidth	509
Configuring and Verifying BGP Link Bandwidth	509
Configuration Examples for BGP Link Bandwidth	511
Example: BGP Link Bandwidth Configuration	511
Example: Verifying BGP Link Bandwidth	514
Where to Go Next	515
Additional References	515
Feature Information for BGP Link Bandwidth	516

CHAPTER 26**iBGP Multipath Load Sharing 519**

Finding Feature Information	519
Restrictions for iBGP Multipath Load Sharing	519
Information about iBGP Multipath Load Sharing	520
iBGP Multipath Load Sharing Overview	520
Benefits of iBGP Multipath Load Sharing	521
How to Configure iBGP Multipath Load Sharing	522
Configuring iBGP Multipath Load Sharing	522
Verifying iBGP Multipath Load Sharing	522
Monitoring and Maintaining iBGP Multipath Load Sharing	525
Configuration Examples for iBGP Multipath Load Sharing	525
Example: iBGP Multipath Load Sharing in a Non-MPLS Topology	525
Example: iBGP Multipath Load Sharing in an MPLS VPN Topology	526
Additional References	526
Feature Information for iBGP Multipath Load Sharing	528

CHAPTER 27**BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 529**

Finding Feature Information	529
-----------------------------	-----

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	530
Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	530
Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	530
Multipath Load Sharing Between eBGP and iBGP	530
eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network	531
eBGP and iBGP Multipath Load Sharing With Route Reflectors	532
Benefits of Multipath Load Sharing for Both eBGP and iBGP	532
How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	532
Configuring Multipath Load Sharing for Both eBGP and iBGP	532
Verifying Multipath Load Sharing for Both eBGP and iBGP	534
Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature	535
Example: Configuring eBGP and iBGP Multipath Load Sharing	535
Example: Verifying eBGP and iBGP Multipath Load Sharing	535
Where to Go Next	536
Additional References	536
Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	537

CHAPTER 28
Loadsharing IP Packets Over More Than Six Parallel Paths 539

Finding Feature Information	539
Overview of Loadsharing IP Packets over More Than Six Parallel Paths	539
Additional References	540
Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths	541

CHAPTER 29
BGP Policy Accounting 543

Finding Feature Information	543
Prerequisites	543
Information About BGP Policy Accounting	544
BGP Policy Accounting Overview	544
Benefits of BGP Policy Accounting	545
How to Configure BGP Policy Accounting	545
Specifying the Match Criteria for BGP Policy Accounting	545
Classifying the IP Traffic and Enabling BGP Policy Accounting	546
Verifying BGP Policy Accounting	547

Monitoring and Maintaining BGP Policy Accounting	548
Configuration Examples for BGP Policy Accounting	549
Example: Specifying the Match Criteria for BGP Policy Accounting	549
Example: Classifying the IP Traffic and Enabling BGP Policy Accounting	549
Additional References	550
Feature Information for BGP Policy Accounting	551

CHAPTER 30

BGP Policy Accounting Output Interface Accounting	553
Finding Feature Information	553
Prerequisites for BGP PA Output Interface Accounting	554
Information About BGP PA Output Interface Accounting	554
BGP PA Output Interface Accounting	554
Benefits of BGP PA Output Interface Accounting	555
How to Configure BGP PA Output Interface Accounting	555
Specifying the Match Criteria for BGP PA	555
Classifying the IP Traffic and Enabling BGP PA	557
Verifying BGP Policy Accounting	559
Configuration Examples for BGP PA Output Interface Accounting	562
Example: Specifying the Match Criteria for BGP Policy Accounting	562
Example: Classifying the IP Traffic and Enabling BGP Policy Accounting	562
Additional References	562
Feature Information for BGP Policy Accounting Output Interface Accounting	564

CHAPTER 31

BGP Cost Community	567
Finding Feature Information	567
Prerequisites for the BGP Cost Community Feature	567
Restrictions for the BGP Cost Community Feature	568
Information About the BGP Cost Community Feature	568
BGP Cost Community Overview	568
How the BGP Cost Community Influences the Best Path Selection Process	569
Cost Community Support for Aggregate Routes and Multipaths	569
Influencing Route Preference in a Multi-Exit IGP Network	570
BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links	570
How to Configure the BGP Cost Community Feature	571
Configuring the BGP Cost Community	571

Verifying the Configuration of the BGP Cost Community	573
Troubleshooting Tips	573
Configuration Examples for the BGP Cost Community Feature	574
Example: BGP Cost Community Configuration	574
Example: BGP Cost Community Verification	574
Additional References	575
Feature Information for BGP Cost Community	576

CHAPTER 32
BGP Support for IP Prefix Import from Global Table into a VRF Table 579

Finding Feature Information	579
Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table	580
Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table	580
Information About BGP Support for IP Prefix Import from Global Table into a VRF Table	580
Importing IPv4 Prefixes into a VRF	580
Black Hole Routing	580
Classifying Global Traffic	581
Unicast Reverse Path Forwarding	581
How to Import IP Prefixes from Global Table into a VRF Table	581
Defining IPv4 IP Prefixes to Import	581
Creating the VRF and the Import Route Map	582
Filtering on the Ingress Interface	585
Verifying Global IP Prefix Import	586
Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table	587
Example: Importing IP Prefixes from Global Table into a VRF Table	587
Example: Verifying IP Prefix Import to a VRF Table	588
Additional References	589
Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table	590

CHAPTER 33
BGP Support for IP Prefix Export from a VRF Table into the Global Table 593

Finding Feature Information	593
Information About IP Prefix Export from a VRF Table into the Global Table	594
Benefits of IP Prefix Export from a VRF Table into the Global Table	594
How IP Prefix Export from a VRF Table into the Global Table Works	594

How to Export IP Prefixes from a VRF Table into the Global Table	595
Creating the VRF and the Export Route Map for an Address Family	595
Creating the VRF and the Export Route Map for a VRF (IPv4 only)	598
Displaying Information About IP Prefix Export from a VRF into the Global Table	601
Configuration Examples for IP Prefix Export from a VRF Table into the Global Table	602
Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv6 Address Family	602
Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv4 Address Family	602
Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IP VRF (IPv4 Only)	602
Additional References	603
Feature Information for IP Prefix Export from a VRF Table into the Global Table	603

CHAPTER 34

BGP per Neighbor SoO Configuration	605
Finding Feature Information	605
Prerequisites for BGP per Neighbor SoO Configuration	605
Restrictions for BGP per Neighbor SoO Configuration	606
Information About Configuring BGP per Neighbor SoO	606
Site of Origin BGP Community Attribute	606
Route Distinguisher	606
BGP per Neighbor Site of Origin Configuration	606
Benefits of BGP per Neighbor Site of Origin	607
How to Configure BGP per Neighbor SoO	608
Enabling Cisco Express Forwarding and Configuring VRF Instances	608
Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template	611
BGP Peer Policy Templates	611
Configuring a per Neighbor SoO Value Using a BGP neighbor Command	614
Configuring a per Neighbor SoO Value Using a BGP Peer Group	616
Configuration Examples for BGP per Neighbor SoO Configuration	619
Example: Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template	619
Example: Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template	619
Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command	620

Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command and 4-Byte Autonomous System Numbers 620

Example: Configuring a per Neighbor SoO Value Using a BGP Peer Group 621

Additional References 621

Feature Information for BGP per Neighbor SoO Configuration 622

CHAPTER 35**Per-VRF Assignment of BGP Router ID 623**

Finding Feature Information 623

Prerequisites for Per-VRF Assignment of BGP Router ID 623

Information About Per-VRF Assignment of BGP Router ID 624

BGP Router ID 624

Per-VRF Router ID Assignment 624

Route Distinguisher 624

How to Configure Per-VRF Assignment of BGP Router ID 624

Configuring VRF Instances 624

Associating VRF Instances with Interfaces 626

Manually Configuring a BGP Router ID per VRF 629

Automatically Assigning a BGP Router ID per VRF 634

Configuration Examples for Per-VRF Assignment of BGP Router ID 641

Example: Manually Configuring a BGP Router ID per VRF 641

Example: Automatically Assigning a BGP Router ID per VRF 643

Additional References 647

Feature Information for Per-VRF Assignment of BGP Router ID 648

CHAPTER 36**BGP Next Hop Unchanged 651**

Finding Feature Information 651

Information About BGP Next Hop Unchanged 651

BGP Next Hop Unchanged 651

How to Configure BGP Next Hop Unchanged 652

Configuring the BGP Next Hop Unchanged for an eBGP Peer 652

Configuration Example for BGP Next Hop Unchanged 654

Example: BGP Next Hop Unchanged for an eBGP Peer 654

Additional References 655

Feature Information for BGP Next Hop Unchanged 655

CHAPTER 37

BGP Event-Based VPN Import	657
Finding Feature Information	657
Prerequisites for BGP Event-Based VPN Import	657
Information About BGP Event-Based VPN Import	658
BGP Event-Based VPN Import	658
Import Path Selection Policy	658
Import Path Limit	659
How to Configure BGP Event-Based VPN Import	659
Configuring a Multiprotocol VRF	659
Configuring Event-Based VPN Import Processing for BGP Paths	662
Monitoring and Troubleshooting BGP Event-Based VPN Import Processing	663
Configuration Examples for BGP Event-Based VPN Import	665
Example: Configuring Event-Based VPN Import Processing for BGP Paths	665
Additional References	666
Feature Information for BGP Event-Based VPN Import	666

CHAPTER 38

BGP Best External	669
Finding Feature Information	669
Prerequisites for BGP Best External	670
Restrictions for BGP Best External	670
Information About BGP Best External	671
BGP Best External Overview	671
What the Best External Route Means	671
BGP Best External Feature Operation	671
Configuration Modes for Enabling BGP Best External	672
How to Configure BGP Best External	673
Configuring the BGP Best External Feature	673
Verifying the BGP Best External Feature	675
Configuration Examples for BGP Best External	678
Example: Configuring the BGP Best External Feature	678
Additional References	679
Feature Information for BGP Best External	680

CHAPTER 39

BGP PIC Edge for IP and MPLS-VPN	683
---	------------

Finding Feature Information	683
Prerequisites for BGP PIC	684
Restrictions for BGP PIC	684
Information About BGP PIC	684
Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature	684
How BGP Converges Under Normal Circumstances	685
How BGP PIC Improves Convergence	685
BGP Fast Reroute's Role in the BGP PIC Feature	686
How a Failure Is Detected	687
How BGP PIC Achieves Subsecond Convergence	688
How BGP PIC Improves Upon the Functionality of MPLS VPN–BGP Local Convergence	688
Configuration Modes for Enabling BGP PIC	688
BGP PIC Scenarios	688
IP PE-CE Link and Node Protection on the CE Side (Dual PEs)	688
IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes)	689
IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path	691
IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path	692
Cisco Express Forwarding Recursion	693
How to Configure BGP PIC	694
Configuring BGP PIC	694
Configuration Examples for BGP PIC	697
Example: Configuring BGP PIC	697
Example: Displaying Backup Alternate Paths for BGP PIC	698
Additional References	700
Feature Information for BGP PIC	701

CHAPTER 40**Configuring BGP: RT Constrained Route Distribution 703**

Finding Feature Information	703
Prerequisites for BGP: RT Constrained Route Distribution	704
Restrictions for BGP: RT Constrained Route Distribution	704
Information About BGP: RT Constrained Route Distribution	704
Problem That BGP: RT Constrained Route Distribution Solves	704
Benefits of BGP: RT Constrained Route Distribution	705
BGP RT-Constrain SAFI	706

BGP: RT Constrained Route Distribution Operation	706
RT Constraint NLRI Prefix	707
RT Constrained Route Distribution Process	707
Default RT Filter	708
How to Configure RT Constrained Route Distribution	708
Configuring Multiprotocol BGP on Provider Edge (PE) Routers and Route Reflectors	708
Troubleshooting Tips	710
Connecting the MPLS VPN Customers	710
Defining VRFs on PE Routers to Enable Customer Connectivity	710
Configuring VRF Interfaces on PE Routers for Each VPN Customer	712
Configuring BGP as the Routing Protocol Between the PE and CE Routers	713
Configuring RT Constraint on the PE	715
Configuring RT Constraint on the RR	716
Configuration Examples for BGP: RT Constrained Route Distribution	719
Example: BGP RT Constrained Route Distribution Between a PE and RR	719
Additional References	720
Feature Information for BGP: RT Constrained Route Distribution	722

CHAPTER 41**Configuring BGP Consistency Checker 723**

Finding Feature Information	723
Information About BGP Consistency Checker	723
BGP Consistency Checker	723
How to Configure BGP Consistency Checker	724
Configure BGP Consistency Checker	724
Configuration Examples for BGP Consistency Checker	726
Example: Configuring BGP Consistency Checker	726
Additional References	726
Feature Information for BGP Consistency Checker	727

CHAPTER 42**BGP MIB Support 729**

Finding Feature Information	729
Information About BGP MIB Support	729
BGP MIB Support	729
How to Enable BGP MIB Support	732
Enabling BGP MIB Support	732

Configuration Examples for BGP MIB Support 733

Example: Enabling BGP MIB Support 733

Additional References 733

Feature Information for BGP MIB Support 734

CHAPTER 43**Cisco-BGP-MIBv2 735**

Finding Feature Information 735

Information About Cisco-BGP-MIBv2 735

Benefits of Cisco-BGP-MIBv2 735

Cisco-BGP-MIBv2 Read-Only Objects 736

Cisco-BGP-MIBv2 Trap Objects 738

How to Configure Cisco-BGP-MIBv2 740

Enabling Cisco-BGP-MIBv2 Traps 740

Configuration Examples for Cisco-BGP-MIBv2 741

Example: Enabling Cisco-BGP-MIBv2 741

Additional References 741

Feature Information for Cisco-BGP-MIBv2 742

CHAPTER 44**BGP Additional Paths 743**

Finding Feature Information 743

Information About BGP Additional Paths 743

Problem That Additional Paths Can Solve 743

Benefits of BGP Additional Paths 746

BGP Additional Paths Functionality 746

How to Configure BGP Additional Paths 748

Configuring Additional Paths per Address Family 748

Configuring Additional Paths per Neighbor 750

Configuring Additional Paths Using a Peer Policy Template 752

Filtering and Setting Actions for Additional Paths 754

Displaying Additional Path Information 756

Disabling Additional Paths per Neighbor 757

Configuration Examples for BGP Additional Paths 759

Example: BGP Additional Path Send and Receive Capabilities 759

Example: BGP Additional Paths 759

Example: Neighbor Capabilities Override Address Family Capabilities 760

Example: BGP Additional Paths Using a Peer Policy Template 760
 Additional References 761
 Feature Information for BGP Additional Paths 761

CHAPTER 45**BGP Attribute Filter and Enhanced Attribute Error Handling 763**

Finding Feature Information 763
 Information About BGP Attribute Filtering 763
 BGP Attribute Filter and Enhanced Attribute Error Handling 763
 How to Filter BGP Path Attributes 765
 Treat-as-Withdraw BGP Updates Containing a Specified Path Attribute 765
 Discarding Specific Path Attributes from an Update Message 766
 Displaying Withdrawn or Discarded Path Attributes 767
 Configuration Examples for BGP Attribute Filter 768
 Examples: Withdraw Updates Based on Path Attribute 768
 Examples: Discard Path Attributes from Updates 769
 Additional References 769
 Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling 770

CHAPTER 46**BGP—Support for iBGP Local-AS 773**

Finding Feature Information 773
 Restrictions for Support for iBGP Local-AS 774
 Information About Support for iBGP Local-AS 774
 Support for iBGP Local-AS 774
 Benefits of iBGP Local-AS 775
 How to Configure iBGP Local-AS 776
 Configuring iBGP Local-AS 776
 Configuration Examples for iBGP Local-AS 779
 Example: Configuring iBGP Local-AS 779
 Additional References for Support for iBGP Local-AS 780
 Feature Information for BGP—Support for iBGP Local-AS 780

CHAPTER 47**BGP-Multiple Cluster IDs 783**

Finding Feature Information 783
 Information About BGP-Multiple Cluster IDs 784
 Benefit of Multiple Cluster IDs Per Route Reflector 784

How a CLUSTER_LIST Attribute is Used	784
Behaviors When Disabling Client-to-Client Route Reflection	785
How to Use BGP-Multiple Cluster IDs	787
Configuring a Cluster ID per Neighbor	787
Disabling Intracluster and Intercluster Client-to-Client Reflection	788
Disabling Intracluster Client-to-Client Reflection for Any Cluster ID	789
Disabling Intracluster Client-to-Client Reflection for Specified Cluster IDs	791
Configuration Examples for BGP-Multiple Cluster IDs	792
Example: Per-Neighbor Cluster ID	792
Example: Disabling Client-to-Client Reflection	792
Additional References	793
Feature Information for BGP-Multiple Cluster IDs	794

CHAPTER 48**BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard 797**

Finding Feature Information	797
Restrictions for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard	798
Information About BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard	798
Benefits of RT and VPN Distinguisher Attribute Mapping Range	798
How to Map RTs to RTs Using a Range	798
Replacing an RT with a Range of RTs	798
Replacing a Range of RTs with an RT	802
Configuration Examples for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard	805
Example: Replacing an RT with a Range of RTs	805
Example: Replacing an RT with a Range of VPN Distinguishers	806
Additional References for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard	807
Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard	807

CHAPTER 49**BGP-VPN Distinguisher Attribute 809**

Finding Feature Information	809
Information About BGP-VPN Distinguisher Attribute	809
Role and Benefit of the VPN Distinguisher Attribute	809
How the VPN Distinguisher Attribute Works	810
How to Configure BGP-VPN Distinguisher Attribute	811
Replacing an RT with a VPN Distinguisher Attribute	811
Replacing a VPN Distinguisher Attribute with an RT	815

Configuration Examples for BGP-VPN Distinguisher Attribute	818
Example: Translating RT to VPN Distinguisher to RT	818
Additional References	819
Feature Information for BGP-VPN Distinguisher Attribute	820

CHAPTER 50**BGP-VRF-Aware Conditional Advertisement 821**

Finding Feature Information	821
Information About BGP VRF-Aware Conditional Advertisement	821
VRF-Aware Conditional Advertisement	821
How to Configure BGP VRF-Aware Conditional Advertisement	823
Configuring BGP VRF-Aware Conditional Advertisement	823
Configuration Examples for BGP VRF-Aware Conditional Advertisement	825
Example: Configuring BGP VRF-Aware Conditional Advertisement	825
Example: Verifying BGP VRF-Aware Conditional Advertisement	827
Additional References for BGP VRF-Aware Conditional Advertisement	829
Feature Information for BGP VRF-Aware Conditional Advertisement	830

CHAPTER 51**BGP Diverse Path Using a Diverse-Path Route Reflector 831**

Finding Feature Information	831
Prerequisites for BGP Diverse Path Using a Diverse-Path Route Reflector	832
Restrictions for BGP Diverse Path Using a Diverse-Path Route Reflector	832
Information About BGP Diverse Path Using a Diverse-Path Reflector	832
Limitation that a BGP Diverse Path Overcomes	832
BGP Diverse Path Using a Diverse-Path Route Reflector	833
Triggers to Compute a BGP Diverse Path	834
IGP Metric Check	835
Route Reflector Determination	835
How to Configure a BGP Diverse-Path Route Reflector	835
Determining Whether You Need to Disable the IGP Metric Check	835
Configuring the Route Reflector for BGP Diverse Path	836
Configuration Examples for BGP Diverse Path Using a Diverse-Path Route Reflector	839
Example: Configuring BGP Diverse Path Where Additional Path Is the Backup Path	839
Example: Configuring BGP Diverse Path Where Additional Path Is the Multipath	840
Example: Configuring BGP Diverse Path Where Both Multipath and Backup Path Calculations Are Triggered	840

Example: Configuring Triggering Computation and Installation of a Backup Path	841
Additional References	841
Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector	842

CHAPTER 52**BGP Graceful Shutdown 845**

Finding Feature Information	845
Information About BGP Graceful Shutdown	845
Purpose and Benefits of BGP Graceful Shutdown	845
GSHUT Community	846
BGP GSHUT Enhancement	846
How to Configure BGP Graceful Shutdown	846
Shutting Down a BGP Link Gracefully	846
Filtering BGP Routes Based on the GSHUT Community	848
Configuring BGP GSHUT Enhancement	851
Configuration Examples for BGP Graceful Shutdown	852
Example: Shutting Down a BGP Link Gracefully	852
Example: Filtering BGP Routes Based on the GSHUT Community	853
Example: BGP GSHUT Enhancement	853
Additional References	854
Feature Information for BGP Graceful Shutdown	855



CHAPTER

1

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 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 software.

- [Finding Feature Information, page 1](#)
- [Prerequisites for Cisco BGP, page 1](#)
- [Restrictions for Cisco BGP, page 2](#)
- [Information About Cisco BGP, page 2](#)
- [Additional References, page 16](#)
- [Feature Information for Cisco BGP Overview, page 17](#)

Finding Feature Information

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

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

Prerequisites for Cisco BGP

This document assumes knowledge of CLNS, 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 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

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 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).

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” chapter.

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 Version 4 Functional Overview

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; 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 software supports BGP version 4 and it is this version that has been used by Internet service providers (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. 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, IPv6, and CLNS.

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.

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 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 traveled 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.

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-known discretionary, and optional transitive attributes that are used in BGP best path analysis. Cisco 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 “BGP 4 Prefix Filter and Inbound Route Maps” module and the “BGP Prefix-Based Outbound Route Filtering” module.

BGP uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, the oldest paths are selected as multipaths.

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.

**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 Autonomous Systems

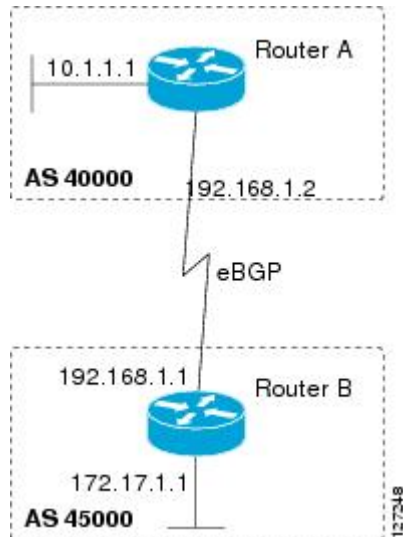
An autonomous system is a network controlled by a single technical administration entity. BGP autonomous systems are used to divide global external networks into individual routing domains where local routing policies are applied. This organization simplifies routing domain administration and simplifies 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 administered 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 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

In Cisco IOS Release 12.0(32)S12, 12.4(24)T, 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 images where only asdot formatting is available.

Table 1: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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 2: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 3: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 12.4(24)T, 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.



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/>.

Classless Interdomain Routing

BGP version 4 supports classless interdomain routing (CIDR). CIDR eliminates classful network boundaries, providing more efficient usage of the IPv4 address space. CIDR provides a method to reduce the size of routing tables by configuring aggregate routes (or supernets). CIDR processes a prefix as an IP address and bit mask (bits are processed from left to right) to define each network. A prefix can represent a network, subnetwork, supernet, or single host route.

For example, using classful IP addressing, the IP address 192.168.2.1 is defined as a single host in the Class C network 192.168.2.0. Using CIDR, the IP address can be shown as 192.168.2.1/16, which defines a network (or supernet) of 192.168.0.0.

CIDR is enabled by default for all routing protocols in Cisco software. Enabling CIDR affects how packets are forwarded, but it does not change the operation of BGP.

Multiprotocol BGP

Cisco software supports multiprotocol BGP extensions as defined in RFC 2858, *Multiprotocol Extensions for BGP-4*. The extensions introduced in this RFC allow BGP to carry routing information for multiple network-layer protocols, including CLNS, 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 that is 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, which allows you more control over your network and resources.

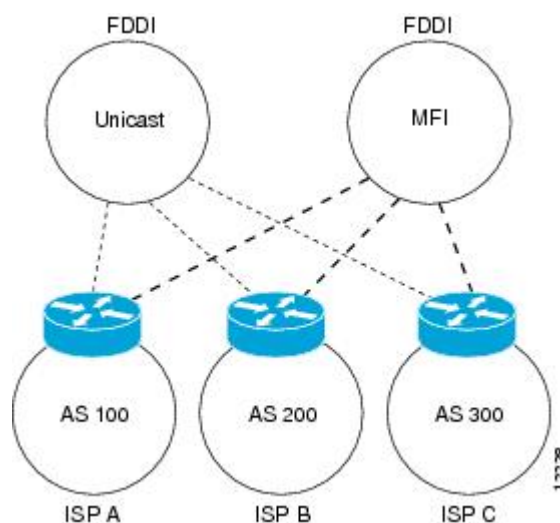
In BGP, the only way to perform interdomain multicast routing is to use the BGP infrastructure that is in place for unicast routing. If the routers are not multicast-capable, or if there are differing policies about where multicast traffic should flow, multicast routing cannot be supported without multiprotocol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



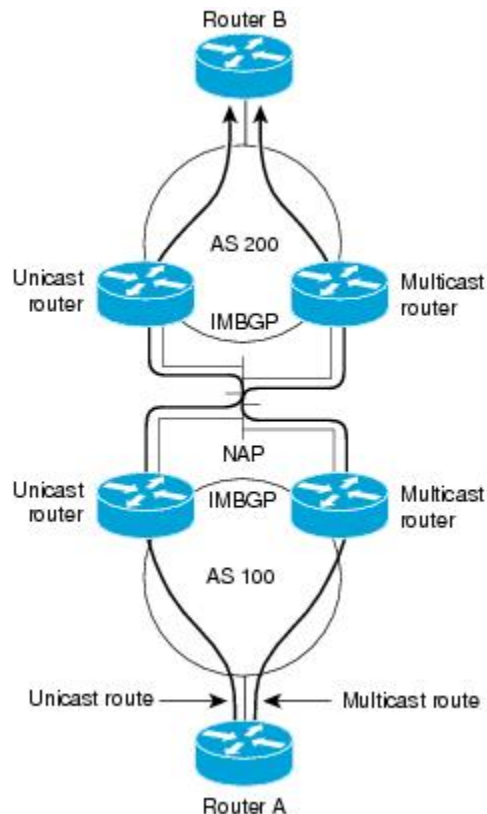
The figure below is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they do not support or are not configured to perform multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both 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 (labeled “IMBGP” 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 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.

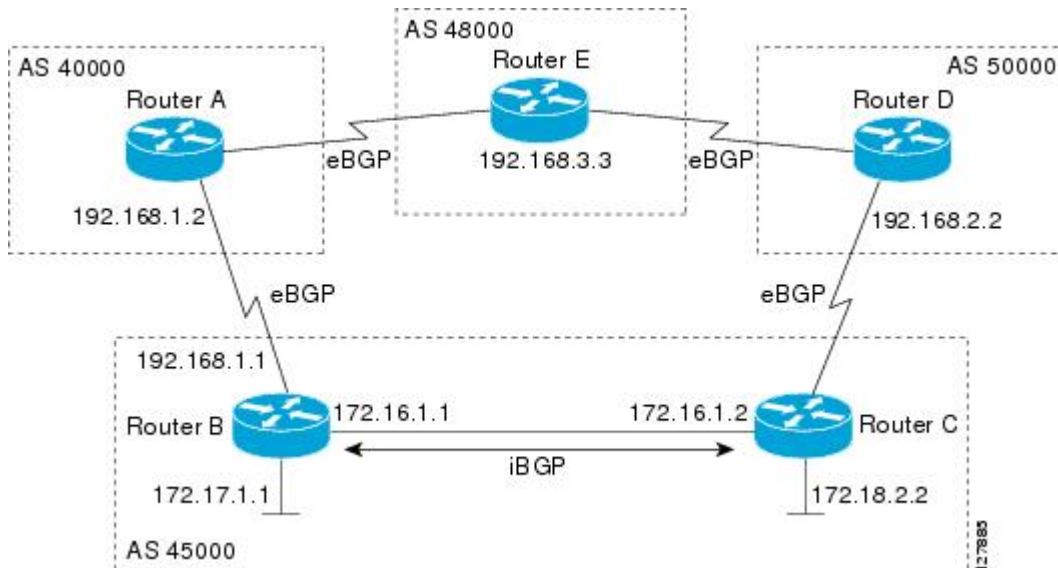
Using the BGP hybrid CLI feature, you can configure commands in the address family VPNv4 format and save these command configurations without modifying an existing NLRI formatted configuration. If you want to use other address family configurations such as IPv4 unicast or multicast, then you must upgrade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

For more details about using BGP hybrid CLI commands, see the "Configuring a Basic BGP Network" module. See the "Multiprotocol BGP" and "Cisco BGP Address Family Model" sections 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



The Cisco BGP AFI model introduced new command-line interface (CLI) commands supported by a new internal structure. 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 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 four address families in Cisco IOS software; IPv4, IPv6, CLNS, and VPNv4. In Cisco IOS Release 12.2(33)SRB, and later releases, support for the L2VPN address family was introduced, and 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 software. To ensure compatibility between networks running all types of AFI and SAFI configuration, we recommend configuring BGP on Cisco IOS devices using the multiprotocol BGP address family model.

Table 4: SAFIs Supported by Cisco IOS Software

SAFI Field Value	Description	Reference
1	NLRI used for unicast forwarding.	RFC 2858
2	NLRI used for multicast forwarding.	RFC 2858

SAFI Field Value	Description	Reference
3	NLRI used for both unicast and multicast forwarding.	RFC 2858
4	NLRI with MPLS labels.	RFC 3107
64	Tunnel SAFI.	draft-nalawade-kapoor-tunnel-safi-01.txt
65	Virtual Private LAN Service (VPLS).	—
66	BGP MDT SAFI.	draft-nalawade-idr-mdt-safi-00.txt
128	MPLS-labeled VPN address.	RFC-ietf-13vpn-rfc2547bis-03.txt

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.

In Cisco IOS Release 12.0(28)S, 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.

In Cisco IOS Release 12.0(29)S, the multicast distribution tree (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 (CEF) 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 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

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.

Under L2VPN address family the following BGP command-line interface (CLI) commands are supported:

- **bgp scan-time**
- **bgp nexthop**
- **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 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.

For details on configuring BGP under the L2VPN address family, see the “BGP Support for the L2VPN Address Family” module.

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:

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

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

```
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
    neighbor 192.168.3.2 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:

```
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 behavior 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:

```
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.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1700	<i>Assigned Numbers</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 3107	<i>Carrying Label Information in BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 4893	<i>BGP Support for Four-Octet AS Number Space</i>

RFC	Title
RFC 5396	<i>Textual Representation of Autonomous System (AS) Numbers</i>
RFC 5398	<i>Autonomous System (AS) Number Reservation for Documentation Use</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 . An account on Cisco.com is not required.

Table 5: Feature Information for Cisco BGP Overview

Feature Name	Releases	Feature Information
Multiprotocol BGP	Cisco IOS XE 3.1.0SG	<p>Cisco IOS software supports multiprotocol BGP extensions as defined in RFC 2858, <i>Multiprotocol Extensions for BGP-4</i>. The extensions introduced in this RFC allow BGP to carry routing information for multiple network layer protocols including CLNS, 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.</p>



CHAPTER 2

BGP 4

BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems).

- [Finding Feature Information, page 19](#)
- [Information About BGP 4, page 19](#)
- [How to Configure BGP 4, page 25](#)
- [Configuration Examples for BGP 4, page 61](#)
- [Additional References, page 66](#)
- [Feature Information for BGP 4, page 67](#)

Finding Feature Information

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

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

Information About BGP 4

BGP Version 4 Functional Overview

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; 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 software supports BGP version 4 and it is this version that has been used by Internet service providers (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. 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, IPv6, and CLNS.

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.

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 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 traveled 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.

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-known discretionary, and optional transitive attributes that are used in BGP best path analysis. Cisco 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 “BGP 4 Prefix Filter and Inbound Route Maps” module and the “BGP Prefix-Based Outbound Route Filtering” module.

BGP uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, the oldest paths are selected as multipaths.

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.

**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 software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the device, the software chooses the highest IPv4 address configured on a physical interface of the device 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 device does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking devices. A peer device is a BGP-speaking device that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a neighbor, but because this can imply the idea that the BGP devices are directly connected with no other device in between, the term *neighbor* will be avoided whenever possible in this document. A BGP speaker is the local device, 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 that is controlled by a single technical administration entity. Peer devices 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.

BGP Peer Session Establishment

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

- **Idle**—The initial state that the BGP routing process enters when the routing process is enabled or when the device is reset. In this state, the device waits for a start event, such as a peering configuration with a remote peer. After the device receives a TCP connection request from a remote peer, the device initiates another start event to wait for a timer before starting a TCP connection to a remote peer. If the device is reset, 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 device 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.

BGP Session Reset

Whenever the routing policy changes due to a configuration change, BGP peering sessions must be reset by using the **clear ip bgp** command. Cisco 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 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 device 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 devices must support the route refresh capability. To determine if a BGP device supports this capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the device 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 Generating AS_SET Information

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 a 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 device to take effect. Performing outbound reset causes the new local outbound policy configured on the device 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 device or an outbound reset on the peer device. Outbound policy changes require an outbound reset on the local device or an inbound reset on the peer device.

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

Table 6: Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead.	The prefixes in the BGP, IP, and Forwarding Information Base (FIB) tables provided by the neighbor are lost. A hard reset is not recommended.
Outbound soft reset	No configuration, and no storing of routing table updates.	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session and cache. Does not require storing of routing table updates, and has no memory overhead.	Both BGP devices must support the route refresh capability. Note Does not reset outbound routing table updates.

Type of Reset	Advantages	Disadvantages
Configured inbound soft reset (uses the neighbor soft-reconfiguration router configuration command)	<p>Can be used when both BGP devices do not support the automatic route refresh capability.</p> <p>The bgp soft-reconfig-backup command was introduced to configure inbound soft reconfiguration for peers that do not support the route refresh capability.</p>	<p>Requires preconfiguration.</p> <p>Stores all received (inbound) routing policy updates without modification; is memory-intensive.</p> <p>Recommended only when absolutely necessary, such as when both BGP devices do not support the automatic route refresh capability.</p> <p>Note Does not reset outbound routing table updates.</p>

Once you have defined two devices 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 if you make a similar configuration change, you must reset BGP connections in order for the configuration change to take effect.

A soft reset updates the routing table for inbound and outbound routing updates. Cisco software supports soft reset without any prior configuration. This soft reset allows the dynamic exchange of route refresh requests and routing information between BGP devices, and allows 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.

BGP Peer Groups

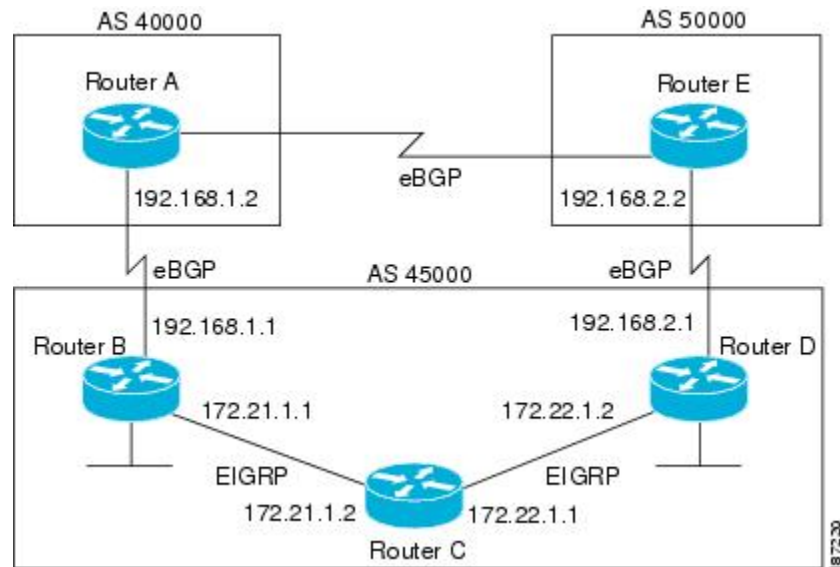
Often, in a BGP network, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into BGP peer groups to simplify configuration and, more importantly, 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 devices using eBGP to communicate to a number of different autonomous systems, using eBGP to communicate between the two border devices 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 5: BGP Backdoor Route Topology



How to Configure BGP 4

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.

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.



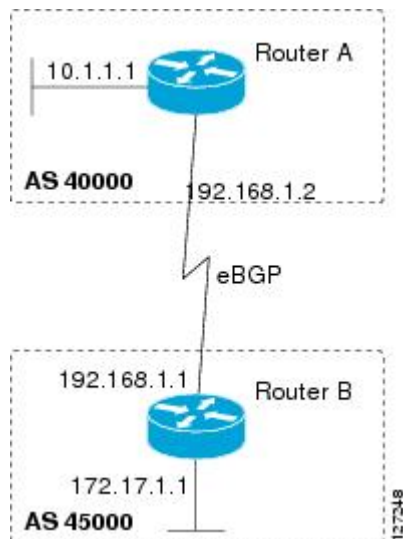
Note

A device that runs Cisco 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 devices. 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 6: 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 40000</pre>	<p>Configures a BGP routing process, and enters router configuration mode for the specified routing process.</p> <ul style="list-style-type: none"> Use the <i>autonomous-system-number</i> argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	<p>network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# network 10.1.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> For exterior protocols, the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 5	<p>bgp router-id <i>ip-address</i></p> <p>Example:</p> <pre>Device(config-router)# bgp router-id 10.1.1.99</pre>	<p>(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP.</p> <ul style="list-style-type: none"> Use the <i>ip-address</i> argument to specify a unique router ID within the network. <p>Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.</p>
Step 6	<p>timers bgp <i>keepalive holdtime</i></p> <p>Example:</p> <pre>Device(config-router)# timers bgp 70 120</pre>	<p>(Optional) Sets BGP network timers.</p> <ul style="list-style-type: none"> Use the <i>keepalive</i> 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 <i>holdtime</i> argument to specify the interval, in seconds, after which the software, having not received a keepalive message, declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds.
Step 7	<p>bgp fast-external-fallover</p> <p>Example:</p> <pre>Device(config-router)# bgp fast-external-fallover</pre>	<p>(Optional) Enables the automatic resetting of BGP sessions.</p> <ul style="list-style-type: none"> By default, the BGP sessions of any directly adjacent external peers are reset if the link used to reach them goes down.

	Command or Action	Purpose
Step 8	bgp log-neighbor-changes Example: <pre>Device(config-router)# bgp log-neighbor-changes</pre>	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. <ul style="list-style-type: none"> • 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.
Step 9	end Example: <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 10	show ip bgp [network] [network-mask] Example: <pre>Device# show ip bgp</pre>	(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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .

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 devices (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 neighboring devices that are to be BGP peers.

Before You Begin

Before you perform this task, perform the “Configuring a BGP Routing Process” 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# 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 device.

	Command or Action	Purpose
Step 5	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: <pre>Device(config-router)# address-family ipv4 unicast</pre>	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	neighbor <i>ip-address</i> activate Example: <pre>Device(config-router-af)# neighbor 192.168.1.1 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.
Step 7	end Example: <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	show ip bgp [<i>network</i>] [<i>network-mask</i>] Example: <pre>Device# show ip bgp</pre>	(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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 9	show ip bgp neighbors [<i>neighbor-address</i>] Example: <pre>Device(config-router-af)# show ip bgp neighbors 192.168.2.2</pre>	(Optional) Displays information about the TCP and BGP connections to neighbors. Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i> .

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

                Sent          Rcvd
Prefix activity: ----          ----
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
                                Outbound  Inbound
Local Policy Denied Prefixes:  -----
AS_PATH loop:          n/a            1
Bestpath from this peer: 1             n/a
Total:                  1             1
Number of NLRI 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
Connection is ECN Disabled
Local host: 192.168.1.2, Local port: 179
Foreign host: 192.168.1.1, Foreign port: 37725
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x12F4F2C):
Timer      Starts  Wakeups          Next
Retrans    14      0             0x0
TimeWait   0        0             0x0
AckHold    13      8             0x0
SendWnd    0        0             0x0
KeepAlive  0        0             0x0
GiveUp     0        0             0x0
PmtuAger   0        0             0x0
DeadWait   0        0             0x0
iss: 165379618 snduna: 165379963 sndnxt: 165379963 sndwnd: 16040
irs: 3127821601 rcvnx: 3127821993 rcvwnd: 15993 delrcvwnd: 391
SRTT: 254 ms, RTTO: 619 ms, RTV: 365 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
Datagrams (max data segment is 1460 bytes):
Rcvd: 20 (out of order: 0), with data: 15, total data bytes: 391
Sent: 22 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 04

```

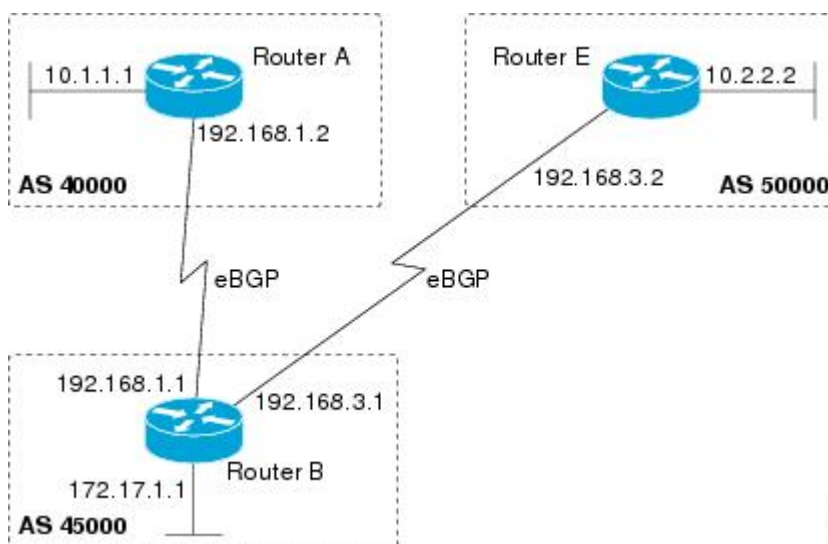
Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP devices.

Configuring a BGP Peer for the IPv4 VRF Address Family

Perform this optional task to configure BGP between two IPv4 devices (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 neighboring devices that are to be BGP IPv4 VRF address family peers.

Figure 7: BGP Topology for IPv4 VRF Address Family



Before You Begin

Before you perform this task, perform the “Configuring a BGP Routing Process” 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example:	Enters interface configuration mode.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding vpn1	Associates a VPN VRF instance with an interface or subinterface.

	Command or Action	Purpose
Step 5	<p>ip address <i>ip-address mask</i> [secondary [vrf <i>vrf-name</i>]]</p> <p>Example:</p> <pre>Device(config-if)# ip address 192.168.3.1 255.255.255.0</pre>	Sets an IP address for an interface.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
Step 7	<p>ip vrf <i>vrf-name</i></p> <p>Example:</p> <pre>Device(config)# ip vrf vpn1</pre>	<p>Configures a VRF routing table and enters VRF configuration mode.</p> <ul style="list-style-type: none"> • Use the <i>vrf-name</i> argument to specify a name to be assigned to the VRF.
Step 8	<p>rd <i>route-distinguisher</i></p> <p>Example:</p> <pre>Device(config-vrf)# rd 45000:5</pre>	<p>Creates routing and forwarding tables and specifies the default route distinguisher for a VPN.</p> <ul style="list-style-type: none"> • Use the <i>route-distinguisher</i> argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 9	<p>route-target {import export both} <i>route-target-ext-community</i></p> <p>Example:</p> <pre>Device(config-vrf)# route-target both 45000:100</pre>	<p>Creates a route target extended community for a VRF.</p> <ul style="list-style-type: none"> • Use the import keyword to import routing information from the target VPN extended community. • Use the export keyword to export routing information to the target VPN extended community. • Use the both keyword to import both import and export routing information to the target VPN extended community. • Use the <i>route-target-ext-community</i> 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.
Step 10	<p>exit</p> <p>Example:</p> <pre>Device(config-vrf)# exit</pre>	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 11	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 12	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 vrf vpn1	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • Use the unicast keyword to specify the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 13	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router-af)# neighbor 192.168.3.2 remote-as 50000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 14	neighbor { <i>ip-address</i> <i>peer-group-name</i> } maximum-prefix <i>maximum</i> [<i>threshold</i>] [restart <i>restart-interval</i>] [warning-only] Example: Device(config-router-af)# neighbor 192.168.3.2 maximum-prefix 10000 warning-only	Controls how many prefixes can be received from a neighbor. <ul style="list-style-type: none"> • Use the <i>maximum</i> 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 device. • Use the <i>threshold</i> argument to specify an integer representing a percentage of the maximum prefix limit at which the device starts to generate a warning message. • Use the warning-only keyword to allow the device to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.
Step 15	neighbor <i>ip-address</i> activate Example: Device(config-router-af)# neighbor 192.168.3.2 activate	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local device.

	Command or Action	Purpose
Step 16	end Example: Device(config-router-af) # end	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

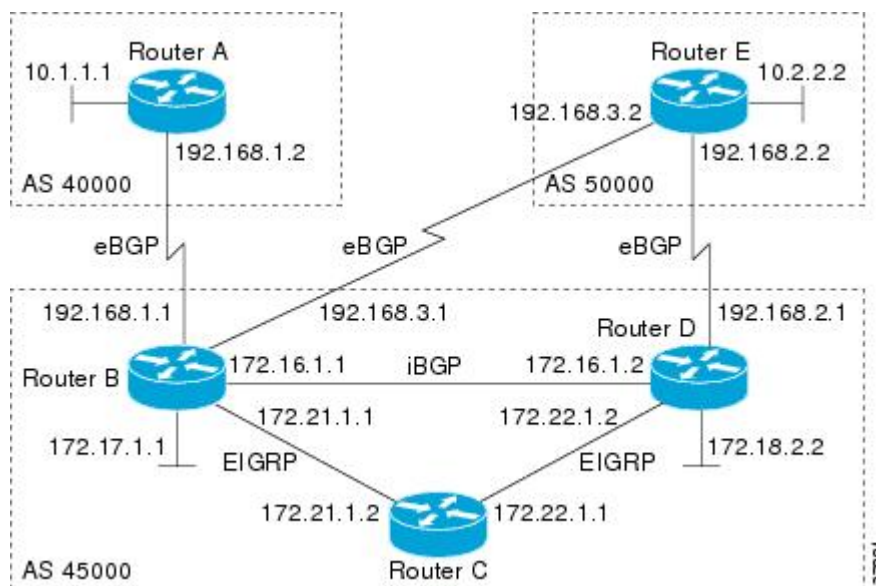
Use the **ping vrf** command to verify basic network connectivity between the BGP devices, and use the **show ip vrf** command to verify that the VRF instance has been created.

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 devices.

Figure 8: BGP Peer Topology



**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. **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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: Device(config-router)# 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.
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.3.2 remote-as 50000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } description <i>text</i> Example: Device(config-router)# neighbor 192.168.3.2 description finance	(Optional) Associates a text description with the specified neighbor.
Step 7	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 multicast	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<ul style="list-style-type: none"> For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.3.2 activate</pre>	Enables the exchange of information with a BGP neighbor.
Step 10	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} advertisement-interval <i>seconds</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25</pre>	(Optional) Sets the minimum interval between the sending of BGP routing updates.
Step 11	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} default-originate [<i>route-map map-name</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.3.2 default-originate</pre>	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 12	<p>exit-address-family</p> <p>Example:</p> <pre>Device(config-router-af)# exit-address-family</pre>	Exits address family configuration mode and enters router configuration mode.
Step 13	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} shutdown</p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 shutdown</pre>	<p>(Optional) Disables a BGP peer or peer group.</p> <p>Note 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.</p>
Step 14	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 15	<p>show ip bgp ipv4 multicast [<i>command</i>]</p> <p>Example:</p> <pre>Device# show ip bgp ipv4 multicast</pre>	<p>(Optional) Displays IPv4 multicast database-related information.</p> <ul style="list-style-type: none"> Use the <i>command</i> argument to specify any multiprotocol BGP command that is supported. To see the supported commands, use the ? prompt on the CLI.

	Command or Action	Purpose
Step 16	show ip bgp neighbors [<i>neighbor-address</i>] [received-routes routes advertised-routes paths <i>regex</i> dampened-routes received prefix-filter] Example: Device# 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 device 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
   Network        Next Hop           Metric LocPrf Weight Path
*> 10.2.2.0/24    192.168.3.2         0         0 50000 i
*> 172.17.1.0/24 0.0.0.0             0         0 32768 i
```

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 holdtimes
Neighbor capabilities:
  Route refresh: advertised and received (old & new)
  Address family IPv4 Unicast: advertised
  Address family IPv4 Multicast: advertised and received
!
For address family: IPv4 Multicast
BGP table version 3, neighbor version 3/0
Output queue size : 0
Index 1, Offset 0, Mask 0x2
1 update-group member
  Uses NEXT_HOP attribute for MBGP NLRIs
Prefix activity:
  Prefixes Current:      1          1 (Consumes 48 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
                               Outbound  Inbound
Local Policy Denied Prefixes:  -----
  Bestpath from this peer:    1          n/a
  Total:                     1          0
Number of NLRIs in the update sent: max 0, min 0
```

```

Minimum time between advertisement runs is 25 seconds
Connections established 8; dropped 7
Last reset 01:48:54, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 192.168.3.1, Local port: 13172
Foreign host: 192.168.3.2, Foreign port: 179
!
```

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” section.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	no route-map map-name Example: Device(config)# no route-map bgp-to-eigrp	Removes a route map from the running configuration. <ul style="list-style-type: none"> • In this example, a route map named bgp-to-eigrp is removed from the configuration.
Step 4	router eigrp autonomous-system-number Example: Device(config)# router eigrp 100	Enters router configuration mode for the specified routing process.
Step 5	no redistribute protocol [as-number] Example: Device(config-router)# no redistribute bgp 45000	Disables the redistribution of routes from one routing domain into another routing domain. <ul style="list-style-type: none"> • In this example, the configuration of the redistribution of BGP routes into the EIGRP routing process is removed from the running configuration. <p>Note If a route map was included in the original redistribute command configuration, remember to remove the route-map command configuration as in Step 3 in this example task.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.
Step 7	show running-config Example: Device# show running-config	(Optional) Displays the current running configuration on the router. <ul style="list-style-type: none"> • Use this command to verify that the redistribute and route-map commands are removed from the router configuration.

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 devices 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

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. Note that the memory requirements for storing the inbound update information can become quite large.

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 inbound soft reconfiguration.
10. **exit**
11. **route-map** *map-name* [permit | deny] [*sequence-number*]
12. **set ip next-hop** *ip-address*
13. **end**
14. **show ip bgp neighbors** [*neighbor-address*]
15. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	bgp log-neighbor-changes Example: Device(config-router)# bgp log-neighbor-changes	Enables logging of BGP neighbor resets.
Step 5	bgp soft-reconfig-backup Example: Device(config-router)# bgp soft-reconfig-backup	Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability. <ul style="list-style-type: none"> • 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.
Step 6	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 7	neighbor {<i>ip-address</i> <i>peer-group-name</i>} soft-reconfiguration [inbound] Example: Device(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound	Configures the Cisco software to start storing updates. <ul style="list-style-type: none"> • 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.
Step 8	neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>map-name</i> {in out}	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> • In this example, the route map named LOCAL will be applied to incoming routes.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 route-map LOCAL in</pre>	
Step 9	Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.	—
Step 10	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 11	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map LOCAL permit 10</pre>	<p>Configures a route map and enters route-map configuration mode.</p> <ul style="list-style-type: none"> In this example, a route map named LOCAL is created.
Step 12	<p>set ip next-hop <i>ip-address</i></p> <p>Example:</p> <pre>Device(config-route-map)# set ip next-hop 192.168.1.144</pre>	<p>Specifies where output packets that pass a match clause of a route map for policy routing.</p> <ul style="list-style-type: none"> In this example, the ip address is set to 192.168.1.144.
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	<p>show ip bgp neighbors [<i>neighbor-address</i>]</p> <p>Example:</p> <pre>Device# show ip bgp neighbors 192.168.1.2</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 15	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>]</p> <p>Example:</p> <pre>Device# show ip bgp</pre>	<p>(Optional) Displays the entries in the BGP routing table.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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 0, 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** *regex* | **dampened-routes** | **received** *prefix-filter*]
5. **show ip bgp paths**
6. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>clear ip bgp {* <i>autonomous-system-number</i> <i>neighbor-address</i>} [soft [in out]]</p> <p>Example:</p> <pre>Device# clear ip bgp *</pre>	<p>Clears and resets BGP neighbor sessions:</p> <ul style="list-style-type: none"> • In the example provided, all BGP neighbor sessions are cleared and reset.
Step 3	<p>show ip bgp [<i>network-address</i>] [<i>network-mask</i>] [longer-prefixes] [prefix-list <i>prefix-list-name</i> route-map <i>route-map-name</i>] [shorter prefixes <i>mask-length</i>]</p> <p>Example:</p> <pre>Device# show ip bgp 10.1.1.0 255.255.255.0</pre>	<p>Displays all the entries in the BGP routing table:</p> <ul style="list-style-type: none"> • In the example provided, the BGP routing table information for the 10.1.1.0 network is displayed.
Step 4	<p>show ip bgp neighbors [<i>neighbor-address</i>] [received-routes routes advertised-routes paths <i>regex</i>] [dampened-routes received <i>prefix-filter</i>]</p> <p>Example:</p> <pre>Device# show ip bgp neighbors 192.168.3.2 advertised-routes</pre>	<p>Displays information about the TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> • In the example provided, the routes advertised from the device to BGP neighbor 192.168.3.2 on another device are displayed.
Step 5	<p>show ip bgp paths</p> <p>Example:</p> <pre>Device# show ip bgp paths</pre>	<p>Displays information about all the BGP paths in the database.</p>
Step 6	<p>show ip bgp summary</p> <p>Example:</p> <pre>Device# show ip bgp summary</pre>	<p>Displays information about the status of all BGP connections.</p>

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

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 device 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip route <i>prefix mask {ip-address interface-type interface-number [ip-address]}</i> [<i>distance</i>] [<i>name</i>] [permanent track number] [tag tag] Example: Device(config)# ip route 172.0.0.0 255.0.0.0 null 0	Creates a static route.

	Command or Action	Purpose
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 5	redistribute static Example: Device(config-router)# redistribute static	Redistributes routes into the BGP routing table.
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

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 “BGP Route Aggregation Generating AS_SET Information” section.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	aggregate-address <i>address mask [as-set]</i> Example: Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set	Creates an aggregate entry in a BGP routing table. <ul style="list-style-type: none"> • 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. <p>Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

Suppressing and Unsuppressing the Advertisement of 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	Do one of the following: <ul style="list-style-type: none"> • aggregate-address <i>address mask</i> [summary-only] • aggregate-address <i>address mask</i> [suppress-map <i>map-name</i>] 	Creates an aggregate route. <ul style="list-style-type: none"> • Use the optional summary-only keyword to create the aggregate route (for example, 10.*.*) and also suppresses advertisements of more-specific routes to all neighbors.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only</pre> <p>Example:</p> <pre>Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 suppress-map map1</pre>	<ul style="list-style-type: none"> Use the optional suppress-map keyword to create the aggregate route but suppress advertisement of specified routes. Routes that are suppressed are not advertised to any neighbors. You can use the match clauses of route maps to selectively suppress some more-specific routes of the aggregate and leave others unsuppressed. IP access lists and autonomous system path access lists match clauses are supported. <p>Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>unsuppress-map <i>map-name</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 unsuppress map1</pre>	<p>(Optional) Selectively advertises routes previously suppressed by the aggregate-address command.</p> <ul style="list-style-type: none"> In this example, the routes previously suppressed in Step 5 are advertised to neighbor 192.168.1.2.
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

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, 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, 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 must exist in the BGP routing table in order 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. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **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...*]}
12. **exit**
13. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
14. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	<p>neighbor <i>ip-address</i> advertise-map <i>map-name</i> {exist-map <i>map-name</i> non-exist-map <i>map-name</i>}</p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 advertise-map map1 exist-map map2</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.</p> <ul style="list-style-type: none"> In this example, the prefix (172.17.0.0) matching the ACL in the advertise map (the route map named map1) will be advertised to the neighbor only when a prefix (192.168.50.0) matching the ACL in exist map (the route-map named map2) is in the local BGP table.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 7	<p>route-map <i>map-tag</i> [permit deny] [sequence-number]</p> <p>Example:</p> <pre>Device(config)# route-map map1 permit 10</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, a route map named map1 is created.
Step 8	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 1</pre>	<p>Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.</p> <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by access list 1.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 10	<p>route-map <i>map-tag</i> [permit deny] [sequence-number]</p>	Configures a route map and enters route map configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config)# route-map map2 permit 10</pre>	<ul style="list-style-type: none"> In this example, a route map named map2 is created.
Step 11	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 2</pre>	<p>Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.</p> <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by access list 2.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 13	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log]</p> <p>Example:</p> <pre>Device(config)# access-list 1 permit 172.17.0.0</pre>	<p>Configures a standard access list.</p> <ul style="list-style-type: none"> 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.
Step 14	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log]</p> <p>Example:</p> <pre>Device(config)# access-list 2 permit 192.168.50.0</pre>	<p>Configures a standard access list.</p> <ul style="list-style-type: none"> In this example, access list 2 permits the 192.168.50.0 to be the prefix of the exist-map.
Step 15	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>

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

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 device from using too many system resources. If the device 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 device.

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*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] { deny <i>network / length</i> permit <i>network / length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: Device(config)# ip prefix-list DEFAULT permit 10.1.1.0/24	Configures an IP prefix list. <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
Step 4	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# route-map ROUTE	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named ROUTE is created.
Step 5	match ip address { <i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]} Example: Device(config-route-map)# match ip address prefix-list DEFAULT	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by prefix list DEFAULT.
Step 6	exit Example: Device(config-route-map)# exit	Exits route map configuration mode and enters global configuration mode.
Step 7	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> } default-originate [route-map <i>map-name</i>] Example: Device(config-router)# neighbor 192.168.3.2 default-originate	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 9	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

Originating BGP Routes Using Backdoor Routes

Use this task to indicate to border devices 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 section.

Before You Begin

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 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 172.22.1.2 remote-as 45000	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device. <ul style="list-style-type: none"> • 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.
Step 5	network <i>ip-address</i> backdoor Example: Device(config-router)# network 172.21.1.0 backdoor	Indicates a network that is reachable through a backdoor route.

	Command or Action	Purpose
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>peer-group-name</i> peer-group Example: Device(config-router)# neighbor fingroup peer-group	Creates a BGP peer group.
Step 5	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.1 remote-as 45000	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.
Step 6	neighbor <i>ip-address</i> peer-group <i>peer-group-name</i> Example: Device(config-router)# neighbor 192.168.1.1 peer-group fingroup	Assigns the IP address of a BGP neighbor to a peer group.
Step 7	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 multicast	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. This is the default. • The multicast keyword specifies that IPv4 multicast address prefixes will be exchanged. • The vrf keyword and <i>vrf-name</i> argument specify that IPv4 VRF instance information will be exchanged.

	Command or Action	Purpose
Step 8	<p>neighbor <i>peer-group-name</i> activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor fingroup activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.</p> <p>Note By default, neighbors that are defined using the neighbor remote-as command in router configuration mode exchange only unicast address prefixes. To allow BGP to exchange other address prefix types, such as multicast that is configured in this example, neighbors must also be activated using the neighbor activate command.</p>
Step 9	<p>neighbor <i>ip-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 peer-group fingroup</pre>	<p>Assigns the IP address of a BGP neighbor to a peer group.</p>
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Configuration Examples for BGP 4

Example: Configuring a BGP Process and Customizing Peers

The following example shows the configuration for Router B in the 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

```

Examples: Removing BGP Configuration Commands Using a Redistribution Example

The following examples show first the CLI configuration to enable the redistribution of BGP routes into EIGRP using a route map and then 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 192.168.1.2 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.

Example: Dynamic Inbound Soft Reset

The following example shows the 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
```

Example: Inbound Soft Reset Using Stored Information

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
```

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.

```
Device# 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:

```
Device# 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:

```
Device# 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           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):

```
Device# show ip bgp paths

Address      Hash Refcount Metric Path
0x2FB5DB0   0      5      0 i
0x2FB5C90   1      4      0 i
0x2FB5C00  1361    2      0 50000 i
0x2FB5D20  2625    2      0 40000 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):

```
Device# 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    0 00:03:49      1
192.168.3.2   4 50000   468    467     0    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 configures BGP to not advertise inactive routes:

```
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

The following example configures a maximum route limit in the VRF named RED and configures BGP to not advertise inactive routes through the VRF named RED:

```
Device(config)# ip vrf RED
Device(config-vrf)# rd 50000:10
Device(config-vrf)# maximum routes 1000 10
Device(config-vrf)# exit
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 vrf RED
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

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:

```
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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>
RFC 1774	<i>BGP-4 Protocol Analysis</i>
RFC 1930	<i>Guidelines for Creation, Selection, and Registration on an Autonomous System (AS)</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP 4

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 7: Feature Information for BGP 4

Feature Name	Releases	Feature Information
BGP 4		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 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).



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 implementation of the neighbor and address family commands is explained. This module also contains tasks to configure and customize BGP peers, implement BGP route aggregation, configure BGP route origination, and define BGP backdoor routes. BGP peer group definition is documented, peer session templates are introduced, and update groups are explained,

- [Finding Feature Information, page 69](#)
- [Prerequisites for Configuring a Basic BGP Network, page 70](#)
- [Restrictions for Configuring a Basic BGP Network, page 70](#)
- [Information About Configuring a Basic BGP Network, page 70](#)
- [How to Configure a Basic BGP Network, page 85](#)
- [Configuration Examples for a Basic BGP Network, page 148](#)
- [Where to Go Next, page 161](#)
- [Additional References, page 162](#)
- [Feature Information for Configuring a Basic BGP Network, page 163](#)

Finding Feature Information

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

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

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 device that runs Cisco 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

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 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 software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the device, the software chooses the highest IPv4 address configured on a physical interface of the device 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 device does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking devices. A peer device is a BGP-speaking device that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a neighbor, but because this can imply the idea that the BGP devices are directly connected with no other device in between, the term *neighbor* will be avoided whenever possible in this document. A BGP speaker is the local device, 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 that is controlled by a single technical administration entity. Peer devices 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.

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 Release 12.0(32)S12, 12.4(24)T, 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 images where only asdot formatting is available.

Table 8: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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 9: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 10: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 12.4(24)T, 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.



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/>.

Cisco Implementation of 4-Byte Autonomous System Numbers

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 15.1(1)SG, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain (65538, for example) as the default regular expression match and the output display format for AS numbers. However, 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.

In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, 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.

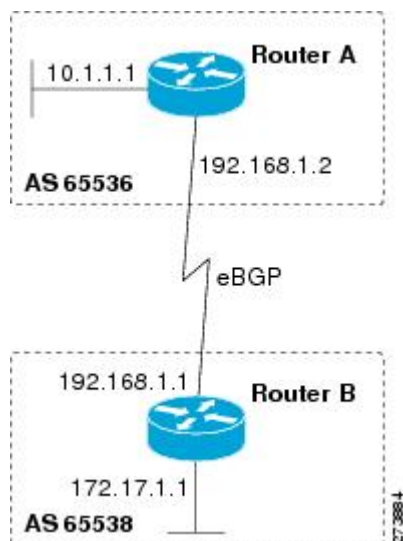
For an example of BGP peers in two autonomous systems using 4-byte numbers, see the figure below. To view a configuration example of the configuration between three neighbor peers in separate 4-byte autonomous systems configured using asdot notation, see the Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers.

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.

**Note**

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 9: 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**—The initial state that the BGP routing process enters when the routing process is enabled or when the device is reset. In this state, the device waits for a start event, such as a peering configuration with a remote peer. After the device receives a TCP connection request from a remote peer, the device initiates another start event to wait for a timer before starting a TCP connection to a remote peer. If the device is reset, 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 device 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 11: Relationships Between BGP Configuration Categories

BGP Configuration Category	Configuration Sets Within Category
Global address family-independent	One set of global address family-independent configurations
Address family-dependent	One set of global address family-dependent configurations per address family

**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
address-family ipv4 unicast vrf <vrf-name>
! 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
neighbor 192.168.3.2 advertisement-interval 25
network 172.16.1.0 mask 255.255.255.0
exit-address-family
address-family ipv4 vrf vpn1
neighbor 192.168.3.2 activate
network 172.21.1.0 mask 255.255.255.0
exit-address-family
```

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 an example of migrating BGP configurations from the NLRI format to the address family format, see the “Example: NLFI to AFI Configuration” section later in this module.

BGP Session Reset

Whenever the routing policy changes due to a configuration change, BGP peering sessions must be reset by using the **clear ip bgp** command. Cisco 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 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 device 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 devices must support the route refresh capability. To determine if a BGP device supports this capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the device 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 Aggregation Route 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 12: Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead.	The prefixes in the BGP, IP, and Forwarding Information Base (FIB) tables provided by the neighbor are lost. Not recommended.
Outbound soft reset	No configuration, no storing of routing table updates.	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session and cache. Does not require storing of routing table updates, and has no memory overhead.	Both BGP routers must support the route refresh capability (in Cisco IOS Release 12.1 and later releases). Note Does not reset outbound routing table updates.
Configured inbound soft reset (uses the neighbor soft-reconfiguration router configuration command)	Can be used when both BGP routers do not support the automatic route refresh capability. In Cisco IOS Release 12.3(14)T, the bgp soft-reconfig-backup command was introduced to configure inbound soft reconfiguration for peers that do not support the route refresh capability.	Requires preconfiguration. Stores all received (inbound) routing policy updates without modification; is memory-intensive. Recommended only when absolutely necessary, such as when both BGP routers do not support the automatic route refresh capability. Note Does not reset outbound routing table updates.

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 Release 12.1 and later releases support 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. Routers running Cisco IOS releases prior to Release 12.1 do not support the route refresh capability and must clear the BGP session using the **neighbor soft-reconfiguration** router configuration command. Clearing the BGP session in this way will have a negative impact upon network operations and should be used only as a last resort.

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 software provides several methods by which you can originate a prefix into BGP. Prior to the BGP conditional route injection feature, the existing methods included redistribution and using the **network** or **aggregate-address** command. However, 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.



Note

Inject maps and exist maps will only match a single prefix per route map clause. To inject additional prefixes, you must configure additional route map clauses. If multiple prefixes are used, the first prefix matched will be used.

BGP Peer Groups

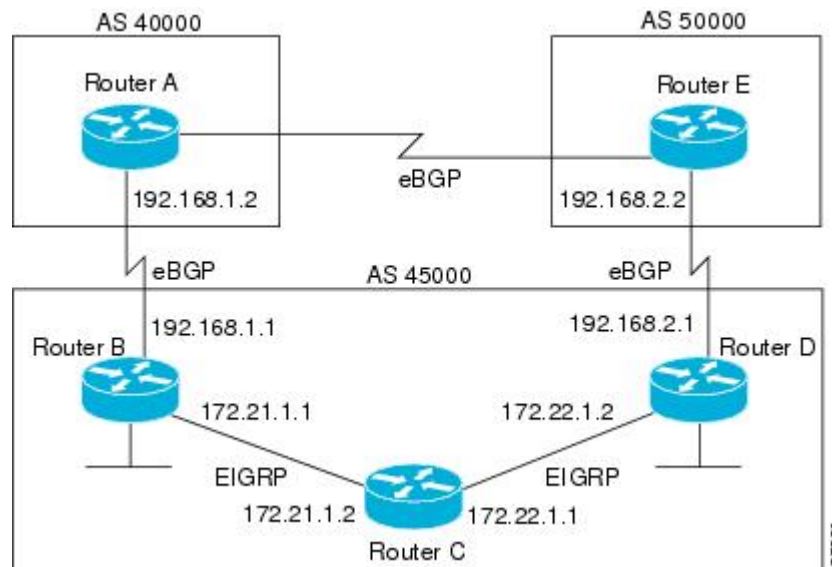
Often, in a BGP network, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update

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 devices using eBGP to communicate to a number of different autonomous systems, using eBGP to communicate between the two border devices 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 10: BGP Backdoor Route Topology



Peer Groups and BGP Update Messages

In Cisco IOS software releases prior to Release 12.0(24)S, 12.2(18)S, or 12.3(4)T, BGP update messages were grouped based on peer group configurations. This method of grouping neighbors for BGP update message generation reduced the amount of system processing resources needed to scan the routing table. This method, however, had the following limitations:

- All neighbors that shared peer group configuration also had to share outbound routing policies.
- All neighbors had to belong to the same peer group and address family. Neighbors configured in different address families could not belong to different peer groups.

These limitations existed to balance optimal update generation and replication against peer group configuration. These limitations could cause the network operator to configure smaller peer groups, which reduced the efficiency of update message generation and limited the scalability of neighbor configuration.

BGP Update Group

The introduction of the BGP (dynamic) update group 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 Release 12.0(24)S, 12.2(18)S, 12.3(4)T, 12.2(27)SBC, and later releases, a new 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.

**Note**

In Cisco IOS Release 12.0(22)S, 12.2(14)S, 12.3(2)T, and prior releases, the update group recalculation delay timer is set to 3 minutes.

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. 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.



Note

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-override**
- **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 a peer session template, 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. Like route maps, inherited peer policy templates are configured with sequence numbers. Also like a route map, an inherited peer policy template is evaluated starting with the **inherit peer-policy** 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 peer-policy** 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 Release 12.2(33)SRE4, 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 Release 12.2(33)SRE4, 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

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

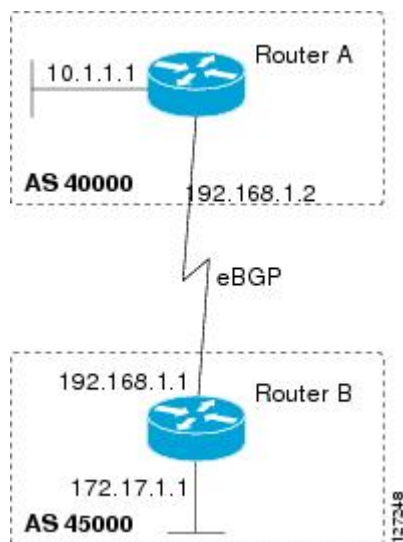
of the features, such as logging neighbor resets and immediate reset of a peer when its link goes down, are enabled by default but are presented here to enhance your understanding of how your BGP network operates.

**Note**

A device that runs Cisco 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 devices. 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 11: 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Configures a BGP routing process, and enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> • Use the <i>autonomous-system-number</i> argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] Example: Device(config-router)# network 10.1.1.0 mask 255.255.255.0	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. <ul style="list-style-type: none"> • For exterior protocols, the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 5	bgp router-id <i>ip-address</i> Example: Device(config-router)# bgp router-id 10.1.1.99	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. <ul style="list-style-type: none"> • Use the <i>ip-address</i> argument to specify a unique router ID within the network. <p>Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.</p>
Step 6	timers bgp <i>keepalive holdtime</i> Example: Device(config-router)# timers bgp 70 120	(Optional) Sets BGP network timers. <ul style="list-style-type: none"> • Use the <i>keepalive</i> 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 <i>holdtime</i> argument to specify the interval, in seconds, after which the software, having not received a keepalive message, declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds.
Step 7	bgp fast-external-fallover	(Optional) Enables the automatic resetting of BGP sessions.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# bgp fast-external-fallover</pre>	<ul style="list-style-type: none"> By default, the BGP sessions of any directly adjacent external peers are reset if the link used to reach them goes down.
Step 8	<p>bgp log-neighbor-changes</p> <p>Example:</p> <pre>Device(config-router)# bgp log-neighbor-changes</pre>	<p>(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets.</p> <ul style="list-style-type: none"> 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.
Step 9	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 10	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>]</p> <p>Example:</p> <pre>Device# show ip bgp</pre>	<p>(Optional) Displays the entries in the BGP routing table.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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 Begin

Before you perform this task, perform the “Configuring a BGP Routing Process” task shown in the prior section.



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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 40000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 remote-as 45000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</p>
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.</p>
Step 7	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
Step 8	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>]</p> <p>Example:</p> <pre>Router# show ip bgp</pre>	<p>(Optional) Displays the entries in the BGP routing table.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 9	<p>show ip bgp neighbors [<i>neighbor-address</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# show ip bgp neighbors 192.168.2.2</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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

          Sent          Rcvd
Prefix activity:
  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

          Outbound      Inbound
Local Policy Denied Prefixes:
  AS_PATH loop:           n/a           1
  Bestpath from this peer: 1           n/a
  Total:                  1           1
Number of NLRI 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
Connection is ECN Disabled
Local host: 192.168.1.2, Local port: 179
Foreign host: 192.168.1.1, Foreign port: 37725
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x12F4F2C):
Timer          Starts      Wakeups      Next
Retrans         14           0           0x0
TimeWait        0           0           0x0
AckHold         13           8           0x0
SendWnd         0           0           0x0
KeepAlive       0           0           0x0
```

```

GiveUp          0          0          0x0
PmtuAger        0          0          0x0
DeadWait        0          0          0x0
iss: 165379618  snduna: 165379963  sndnxt: 165379963  sndwnd: 16040
irs: 3127821601  rcvnxt: 3127821993  rcvwnd: 15993  delrcvwnd: 391
SRTT: 254 ms, RTTO: 619 ms, RTV: 365 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
Datagrams (max data segment is 1460 bytes):
Rcvd: 20 (out of order: 0), with data: 15, total data bytes: 391
Sent: 22 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 04

```

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, on page 99](#). If you do not have BGP peers in a VPN, proceed to the [Customizing a BGP Peer, on page 36](#).

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

Perform this task to configure a Border Gateway Protocol (BGP) routing process and BGP peers when the BGP peers are located in an autonomous system (AS) that uses 4-byte AS 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 (in the “Cisco Implementation of 4-Byte Autonomous System Numbers” section). The 4-byte AS numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in AS number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before You Begin



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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65538	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 65536	Adds the IP address of the neighbor in the specified AS to the IPv4 multiprotocol BGP neighbor table of the local device. • In this example, the 4-byte AS number, 65536, is defined in asplain notation.
Step 5	Repeat Step 4 to define other BGP neighbors, as required.	--
Step 6	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]	Specifies the IPv4 address family and enters address family configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<ul style="list-style-type: none"> The unicast keyword specifies the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	<p>neighbor ip-address activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.2 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.
Step 8	Repeat Step 7 to activate other BGP neighbors, as required.	--
Step 9	<p>network network-number [mask network-mask] [route-map route-map-name]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this AS and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	<p>show ip bgp [network] [network-mask]</p> <p>Example:</p> <pre>Device# show ip bgp 10.1.1.0</pre>	<p>(Optional) Displays the entries in the BGP routing table.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 12	<p>show ip bgp summary</p> <p>Example:</p> <pre>Device# show ip bgp summary</pre>	(Optional) Displays the status of all BGP connections.

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 AS 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
    192.168.1.2 from 192.168.1.2 (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 AS 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:

```
RouterB# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
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
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 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 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  Stated
192.168.1.2     4      65536      6      6        3    0    0 00:01:33      1
```

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 (AS) 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 AS numbers.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ip bgp summary Example: Device# show ip bgp summary	Displays the status of all Border Gateway Protocol (BGP) connections.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65538	Enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> • In this example, the 4-byte AS number, 65538, is defined in asplain notation.

	Command or Action	Purpose
Step 5	bgp asnotation dot Example: <pre>Device(config-router)# bgp asnotation dot</pre>	Changes the default output format of BGP 4-byte AS numbers from asplain (decimal values) to dot notation. Note 4-byte AS 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 6	end Example: <pre>Device(config-router)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	clear ip bgp * Example: <pre>Device# clear ip bgp *</pre>	Clears and resets all current BGP sessions. <ul style="list-style-type: none"> In this example, a hard reset is performed to ensure that the 4-byte AS 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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 8	show ip bgp summary Example: <pre>Device# show ip bgp summary</pre>	Displays the status of all BGP connections.
Step 9	show ip bgp regexp <i>regexp</i> Example: <pre>Device# show ip bgp regexp ^1\.0\$</pre>	Displays routes that match the AS path regular expression. <ul style="list-style-type: none"> In this example, a regular expression to match a 4-byte AS path is configured using asdot format.
Step 10	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 11	router bgp <i>autonomous-system-number</i> Example: <pre>Device(config)# router bgp 65538</pre>	Enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 12	no bgp asnotation dot Example: <pre>Device(config-router)# no bgp asnotation dot</pre>	Resets the default output format of BGP 4-byte AS numbers back to asplain (decimal values). Note 4-byte AS 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.

	Command or Action	Purpose
Step 13	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.
Step 14	clear ip bgp * Example: Device# clear ip bgp *	Clears and resets all current BGP sessions. <ul style="list-style-type: none"> • In this example, a hard reset is performed to ensure that the 4-byte AS 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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .

Examples

The following output from the **show ip bgp summary** command shows the default asplain format of the 4-byte AS numbers. Note the asplain format of the 4-byte AS 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    0 00:03:04    0
192.168.3.2   4      65550    4      4        1    0    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 AS numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 AS 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
192.168.1.2   4      1.0    9      9        1    0    0 00:04:13    0
192.168.3.2   4      1.14   6      6        1    0    0 00:01:24    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 regular expression match format for 4-byte AS paths is changed to asdot notation format. Although a 4-byte AS number can be configured in a regular expression using either asplain format or asdot format, only 4-byte AS 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 AS 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 AS 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.

```
Router# show ip bgp regexp ^65536$
Router# show ip bgp regexp ^1\.0$

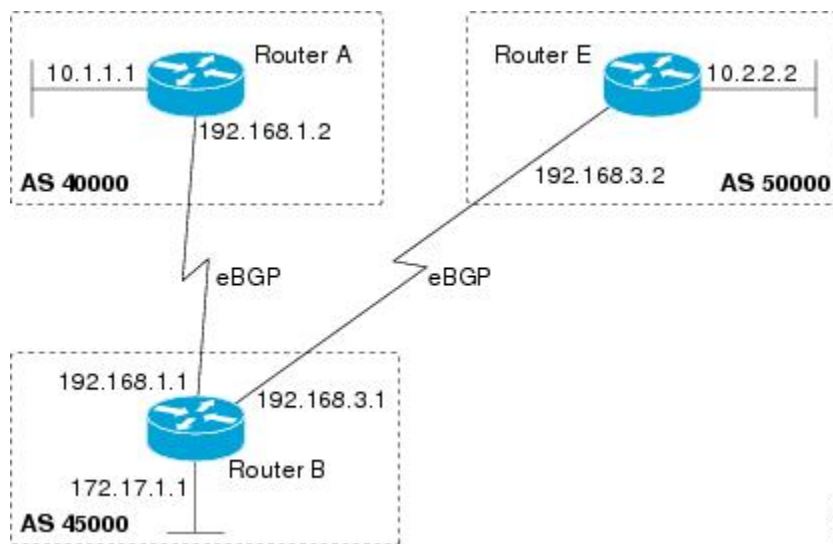
BGP table version is 2, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop          Metric LocPrf Weight Path
*> 10.1.1.0/24    192.168.1.2         0           0 1.0 i
```

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 .

Figure 12: BGP Topology for IPv4 VRF Address Family



Before You Begin

Before you perform this task, perform the [Configuring a BGP Routing Process](#), on page 25 task.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i> Example: Router(config)# ip vrf vpn1	Configures a VRF routing table and enters VRF configuration mode. • Use the <i>vrf-name</i> argument to specify a name to be assigned to the VRF.
Step 4	rd <i>route-distinguisher</i> Example: Router(config-vrf)# rd 45000:5	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the <i>route-distinguisher</i> argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 5	route-target { import export both } <i>route-target-ext-community</i>	Creates a route target extended community for a VRF. • Use the import keyword to import routing information from the target VPN extended community.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-vrf)# route-target both 45000:100</pre>	<ul style="list-style-type: none"> Use the export keyword to export routing information to the target VPN extended community. Use the both keyword to import both import and export routing information to the target VPN extended community. Use the <i>route-target-ext-community</i> 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.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-vrf)# exit</pre>	Exits VRF configuration mode and enters global configuration mode.
Step 7	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 8	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 vrf vpn1</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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 <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 9	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 remote-as 45000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 10	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} maximum-prefix <i>maximum</i> [<i>threshold</i>] [restart <i>restart-interval</i>] [warning-only]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor</pre>	<p>Controls how many prefixes can be received from a neighbor.</p> <ul style="list-style-type: none"> Use the <i>maximum</i> 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.

	Command or Action	Purpose
	<pre>192.168.3.2 maximum-prefix 10000 warning-only</pre>	<ul style="list-style-type: none"> Use the <i>threshold</i> argument to specify an integer representing a percentage of the maximum prefix limit at which the router starts to generate a warning message. Use the warning-only keyword to allow the router to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.
Step 11	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.
Step 12	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

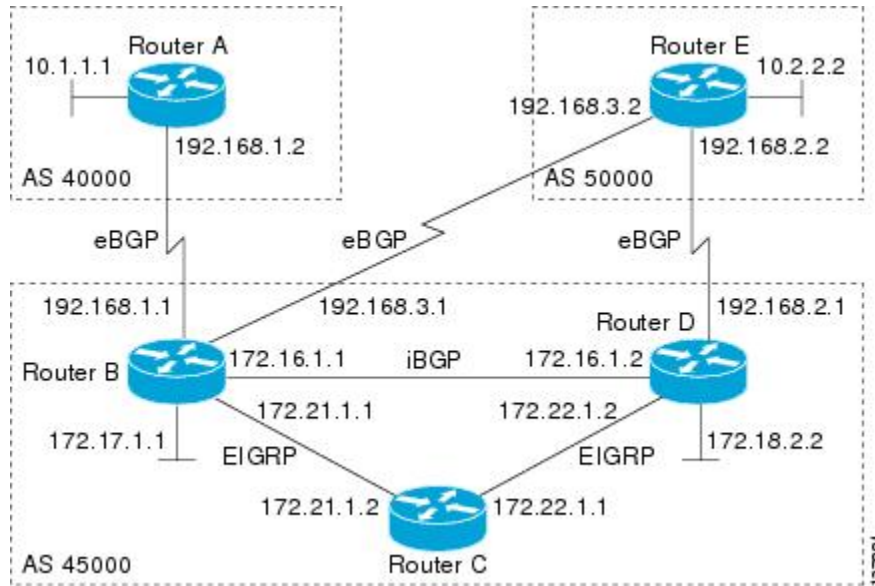
Use the **ping** command to verify basic network connectivity between the BGP routers, and use the **show ip vrf** command to verify that the VRF instance has been created.

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 devices.

Figure 13: BGP Peer Topology



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. **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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process.

	Command or Action	Purpose
		<p>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>
Step 5	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 6	<p>neighbor <i>{ip-address peer-group-name}</i> description <i>text</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 description finance</pre>	(Optional) Associates a text description with the specified neighbor.
Step 7	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 multicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	<p>network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • 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	Purpose
Step 9	neighbor { <i>ip-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.3.2 activate	Enables the exchange of information with a BGP neighbor.
Step 10	neighbor { <i>ip-address</i> <i>peer-group-name</i> } advertisement-interval <i>seconds</i> Example: Device(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25	(Optional) Sets the minimum interval between the sending of BGP routing updates.
Step 11	neighbor { <i>ip-address</i> <i>peer-group-name</i> } default-originate [route-map <i>map-name</i>] Example: Device(config-router-af)# neighbor 192.168.3.2 default-originate	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 12	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and enters router configuration mode.
Step 13	neighbor { <i>ip-address</i> <i>peer-group-name</i> } shutdown Example: Device(config-router)# neighbor 192.168.3.2 shutdown	(Optional) Disables a BGP peer or peer group. Note 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.
Step 14	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.
Step 15	show ip bgp ipv4 multicast [<i>command</i>] Example: Device# show ip bgp ipv4 multicast	(Optional) Displays IPv4 multicast database-related information. <ul style="list-style-type: none"> Use the <i>command</i> 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 [<i>neighbor-address</i>] [received-routes routes advertised-routes paths regexp dampened-routes received prefix-filter]	(Optional) Displays information about the TCP and BGP connections to neighbors.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device# show ip bgp neighbors 192.168.3.2</pre>	

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 device 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
   Network        Next Hop           Metric LocPrf Weight Path
*> 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
```

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 holdtimes
Neighbor capabilities:
  Route refresh: advertised and received (old & new)
  Address family IPv4 Unicast: advertised
  Address family IPv4 Multicast: advertised and received
!
For address family: IPv4 Multicast
BGP table version 3, neighbor version 3/0
Output queue size : 0
Index 1, Offset 0, Mask 0x2
1 update-group member
  Uses NEXT_HOP attribute for MBGP NLRIs
Prefix activity:
  Sent          Rcvd
  ----          -
Prefixes Current:      1          1 (Consumes 48 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
                        Outbound    Inbound
Local Policy Denied Prefixes:  -----
  Bestpath from this peer:          1          n/a
  Total:                            1          0
Number of NLRIs in the update sent: max 0, min 0
Minimum time between advertisement runs is 25 seconds
Connections established 8; dropped 7
Last reset 01:48:54, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
```

```
Local host: 192.168.3.1, Local port: 13172
Foreign host: 192.168.3.2, Foreign port: 179
!
```

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” section.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>no route-map <i>map-name</i></p> <p>Example:</p> <pre>Device(config)# no route-map bgp-to-eigrp</pre>	<p>Removes a route map from the running configuration.</p> <ul style="list-style-type: none"> In this example, a route map named <code>bgp-to-eigrp</code> is removed from the configuration.
Step 4	<p>router eigrp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router eigrp 100</pre>	<p>Enters router configuration mode for the specified routing process.</p>
Step 5	<p>no redistribute <i>protocol</i> [<i>as-number</i>]</p> <p>Example:</p> <pre>Device(config-router)# no redistribute bgp 45000</pre>	<p>Disables the redistribution of routes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> In this example, the configuration of the redistribution of BGP routes into the EIGRP routing process is removed from the running configuration. <p>Note If a route map was included in the original redistribute command configuration, remember to remove the route-map command configuration as in Step 3 in this example task.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>
Step 7	<p>show running-config</p> <p>Example:</p> <pre>Device# show running-config</pre>	<p>(Optional) Displays the current running configuration on the router.</p> <ul style="list-style-type: none"> Use this command to verify that the redistribute and route-map commands are removed from the router configuration.

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

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. Note that the memory requirements for storing the inbound update information can become quite large.

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 inbound soft reconfiguration.
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address*
13. **end**
14. **show ip bgp neighbors** [*neighbor-address*]
15. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	bgp log-neighbor-changes Example: <pre>Device(config-router)# bgp log-neighbor-changes</pre>	Enables logging of BGP neighbor resets.
Step 5	bgp soft-reconfig-backup Example: <pre>Device(config-router)# bgp soft-reconfig-backup</pre>	Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability. <ul style="list-style-type: none"> • 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.
Step 6	neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i> Example: <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 7	neighbor <i>{ip-address peer-group-name}</i> soft-reconfiguration [inbound] Example: <pre>Device(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound</pre>	Configures the Cisco software to start storing updates. <ul style="list-style-type: none"> • 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.
Step 8	neighbor <i>{ip-address peer-group-name}</i> route-map <i>map-name</i> {in out} Example: <pre>Device(config-router)# neighbor 192.168.1.2 route-map LOCAL in</pre>	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> • In this example, the route map named LOCAL will be applied to incoming routes.
Step 9	Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.	—
Step 10	exit Example: <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 11	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# route-map LOCAL permit 10	Configures a route map and enters route-map configuration mode. <ul style="list-style-type: none"> In this example, a route map named LOCAL is created.
Step 12	set ip next-hop <i>ip-address</i> Example: Device(config-route-map)# set ip next-hop 192.168.1.144	Specifies where output packets that pass a match clause of a route map for policy routing. <ul style="list-style-type: none"> In this example, the ip address is set to 192.168.1.144.
Step 13	end Example: Device(config-route-map)# end	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	show ip bgp neighbors [<i>neighbor-address</i>] Example: Device# show ip bgp neighbors 192.168.1.2	(Optional) Displays information about the TCP and BGP connections to neighbors. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 15	show ip bgp [<i>network</i>] [<i>network-mask</i>] Example: Device# show ip bgp	(Optional) Displays the entries in the BGP routing table. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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 0, 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** *regex* | **dampened-routes** | **received** *prefix-filter*]
5. **show ip bgp paths**
6. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	clear ip bgp <i>{* autonomous-system-number neighbor-address}</i> [soft [in out]] Example: Device# clear ip bgp *	Clears and resets BGP neighbor sessions: • In the example provided, all BGP neighbor sessions are cleared and reset.
Step 3	show ip bgp [<i>network-address</i>] [<i>network-mask</i>] [longer-prefixes] [prefix-list <i>prefix-list-name</i> route-map <i>route-map-name</i>] [shorter prefixes <i>mask-length</i>]	Displays all the entries in the BGP routing table: • In the example provided, the BGP routing table information for the 10.1.1.0 network is displayed.

	Command or Action	Purpose
	Example: Device# show ip bgp 10.1.1.0 255.255.255.0	
Step 4	show ip bgp neighbors [<i>neighbor-address</i>] [<i>received-routes</i> <i>routes</i> <i>advertised-routes</i> <i>paths regexp</i> <i>dampened-routes</i> <i>received prefix-filter</i>] Example: Device# show ip bgp neighbors 192.168.3.2 advertised-routes	Displays information about the TCP and BGP connections to neighbors. <ul style="list-style-type: none"> • In the example provided, the routes advertised from the device to BGP neighbor 192.168.3.2 on another device are displayed.
Step 5	show ip bgp paths Example: Device# show ip bgp paths	Displays information about all the BGP paths in the database.
Step 6	show ip bgp summary Example: Device# show ip bgp summary	Displays information about the status of all BGP connections.

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

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 device 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip route <i>prefix mask {ip-address interface-type interface-number [ip-address]} [distance] [name] [permanent track number] [tag tag]</i> Example: Device(config)# ip route 172.0.0.0 255.0.0.0 null 0	Creates a static route.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 5	redistribute static Example: Device(config-router)# redistribute static	Redistributes routes into the BGP routing table.
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

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 “BGP Route Aggregation Generating AS_SET Information” section.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	aggregate-address <i>address mask</i> [as-set] Example: Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set	Creates an aggregate entry in a BGP routing table. <ul style="list-style-type: none"> • 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

	Command or Action	Purpose
		every time the reachability information for the aggregated route changes. Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

Suppressing and Unsuppressing the Advertisement of 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	<p>Do one of the following:</p> <ul style="list-style-type: none"> • aggregate-address <i>address mask</i> [summary-only] • aggregate-address <i>address mask</i> [suppress-map <i>map-name</i>] <p>Example:</p> <pre>Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only</pre> <p>Example:</p> <pre>Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 suppress-map map1</pre>	<p>Creates an aggregate route.</p> <ul style="list-style-type: none"> • Use the optional summary-only keyword to create the aggregate route (for example, 10.*.*.*) and also suppresses advertisements of more-specific routes to all neighbors. • Use the optional suppress-map keyword to create the aggregate route but suppress advertisement of specified routes. Routes that are suppressed are not advertised to any neighbors. You can use the match clauses of route maps to selectively suppress some more-specific routes of the aggregate and leave others unsuppressed. IP access lists and autonomous system path access lists match clauses are supported. <p>Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} unsuppress-map <i>map-name</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 unsuppress map1</pre>	<p>(Optional) Selectively advertises routes previously suppressed by the aggregate-address command.</p> <ul style="list-style-type: none"> • In this example, the routes previously suppressed in Step 5 are advertised to neighbor 192.168.1.2.
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

Suppressing Inactive Route Advertisement Using BGP

Perform this task to suppress the advertisement of inactive routes by BGP. In Cisco IOS Release 12.2(25)S, 12.2(33)SXH, and 15.0(1)M, the **bgp suppress-inactive** command was introduced to configure 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.

Before You Begin

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*] | *vpn**v4* [*unicast*]}
5. **bgp suppress-inactive**
6. **end**
7. **show ip bgp rib-failure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	address-family {ipv4 [mdt multicast unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpnv4 [unicast]} Example: Router(config-router)# address-family ipv4 unicast	Enter address family configuration mode to configure BGP peers to accept address family specific configurations. <ul style="list-style-type: none"> The example creates an IPv4 unicast address family session.
Step 5	bgp suppress-inactive Example: Router(config-router-af)# bgp suppress-inactive	Suppresses BGP advertising of inactive routes. <ul style="list-style-type: none"> BGP advertises inactive routes by default. Entering the no form of this command reenables the advertisement of inactive routes.
Step 6	end Example: Router(config-router-af)# end	Exits address family configuration mode and enters privileged EXEC mode.
Step 7	show ip bgp rib-failure Example: Router# show ip bgp rib-failure	(Optional) Displays BGP routes that are not installed in the RIB.

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.

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

```

Network          Next Hop          RIB-failure      RIB-NH Matches
10.1.15.0/24     10.1.35.5        Higher admin distance  n/a
10.1.16.0/24     10.1.15.1        Higher admin distance  n/a

```

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, 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, 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 must exist in the BGP routing table in order 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. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **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...*]}
12. **exit**
13. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
14. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	neighbor <i>ip-address</i> advertise-map <i>map-name</i> {<i>exist-map map-name</i> <i>non-exist-map map-name</i>} Example: Device(config-router)# neighbor 192.168.1.2 advertise-map map1 exist-map map2	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. <ul style="list-style-type: none"> In this example, the prefix (172.17.0.0) matching the ACL in the advertise map (the route map named map1) will be advertised to the neighbor only when a prefix (192.168.50.0) matching the ACL in exist map (the route-map named map2) is in the local BGP table.
Step 6	exit Example: Device(config-router)# exit	Exits router configuration mode and enters global configuration mode.
Step 7	route-map <i>map-tag</i> [permit deny] [sequence-number] Example: Device(config)# route-map map1 permit 10	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named map1 is created.
Step 8	match ip address {<i>access-list-number</i> [<i>access-list-number</i>... <i>access-list-name</i>...] <i>access-list-name</i> [<i>access-list-number</i>... <i>access-list-name</i>] <i>prefix-list prefix-list-name</i> [<i>prefix-list-name</i>...]}	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by access list 1.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-route-map)# match ip address 1</pre>	
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 10	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map map2 permit 10</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, a route map named map2 is created.
Step 11	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 2</pre>	<p>Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.</p> <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by access list 2.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 13	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log]</p> <p>Example:</p> <pre>Device(config)# access-list 1 permit 172.17.0.0</pre>	<p>Configures a standard access list.</p> <ul style="list-style-type: none"> 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.
Step 14	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log]</p> <p>Example:</p> <pre>Device(config)# access-list 2 permit 192.168.50.0</pre>	<p>Configures a standard access list.</p> <ul style="list-style-type: none"> In this example, access list 2 permits the 192.168.50.0 to be the prefix of the exist-map.

	Command or Action	Purpose
Step 15	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

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

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 device from using too many system resources. If the device 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 device.

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*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] {deny <i>network / length</i> permit <i>network / length</i>} [ge <i>ge-value</i>] [le <i>le-value</i>]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list DEFAULT permit 10.1.1.0/24</pre>	<p>Configures an IP prefix list.</p> <ul style="list-style-type: none"> 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.
Step 4	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map ROUTE</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, a route map named ROUTE is created.
Step 5	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address prefix-list DEFAULT</pre>	<p>Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.</p> <ul style="list-style-type: none"> In this example, the route map is configured to match a prefix permitted by prefix list DEFAULT.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 7	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 40000</pre>	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> } default-originate [<i>route-map map-name</i>] Example: Device(config-router)# neighbor 192.168.3.2 default-originate	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 9	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

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.

Before You Begin

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-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
8. **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...* | *access-list-name...*]
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **set 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...*]}
12. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
13. **exit**
14. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
15. Repeat Step 14 for every prefix list to be created.
16. **exit**
17. **show ip bgp injected-paths**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	bgp inject-map <i>inject-map-name</i> exist-map <i>exist-map-name</i> [copy-attributes]	Specifies the inject map and the exist map for conditional route injection.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH</pre>	<ul style="list-style-type: none"> Use the copy-attributes keyword to specify that the injected route inherit the attributes of the aggregate route.
Step 5	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 6	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map LEARNED_PATH permit 10</pre>	Configures a route map and enters route map configuration mode.
Step 7	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Router(config-route-map)# match ip address prefix-list SOURCE</pre>	<p>Specifies the aggregate route to which a more specific route will be injected.</p> <ul style="list-style-type: none"> In this example, the prefix list named SOURCE is used to redistribute the source of the route.
Step 8	<p>match ip route-source {<i>access-list-number</i> <i>access-list-name</i>} [<i>access-list-number...</i> <i>access-list-name...</i>]</p> <p>Example:</p> <pre>Router(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE</pre>	<p>Specifies the match conditions for redistributing the source of the route.</p> <ul style="list-style-type: none"> In this example, the prefix list named ROUTE_SOURCE is used to redistribute the source of the route. <p>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.</p>
Step 9	<p>exit</p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 10	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map ORIGINATE permit 10</pre>	Configures a route map and enters route map configuration mode.

	Command or Action	Purpose
Step 11	<p>set ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Router(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES</pre>	<p>Specifies the routes to be injected.</p> <ul style="list-style-type: none"> In this example, the prefix list named <code>originated_routes</code> is used to redistribute the source of the route.
Step 12	<p>set community {<i>community-number</i> [additive] [<i>well-known-community</i>] none}</p> <p>Example:</p> <pre>Router(config-route-map)# set community 14616:555 additive</pre>	<p>Sets the BGP community attribute of the injected route.</p>
Step 13	<p>exit</p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 14	<p>ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] {deny <i>network/length</i> permit <i>network/length</i>} [ge <i>ge-value</i>] [le <i>le-value</i>]</p> <p>Example:</p> <pre>Router(config)# ip prefix-list SOURCE permit 10.1.1.0/24</pre>	<p>Configures a prefix list.</p> <ul style="list-style-type: none"> In this example, the prefix list named <code>SOURCE</code> is configured to permit routes from network <code>10.1.1.0/24</code>.
Step 15	<p>Repeat Step 14 for every prefix list to be created.</p>	--
Step 16	<p>exit</p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>
Step 17	<p>show ip bgp injected-paths</p> <p>Example:</p> <pre>Router# show ip bgp injected-paths</pre>	<p>(Optional) Displays information about injected paths.</p>

Examples

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

```
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
*> 172.17.0.0/16 10.0.0.2
```

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 devices 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 section.

Before You Begin

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 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 172.22.1.2 remote-as 45000	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device. <ul style="list-style-type: none"> • 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.
Step 5	network <i>ip-address</i> backdoor Example: Device(config-router)# network 172.21.1.0 backdoor	Indicates a network that is reachable through a backdoor route.
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>router <i>bgp</i> <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 40000</pre>	Enters router configuration mode for the specified routing process.
Step 4	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Device(config-router)# neighbor fingroup peer-group</pre>	Creates a BGP peer group.
Step 5	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.1 remote-as 45000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.
Step 6	<p>neighbor <i>ip-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.1 peer-group fingroup</pre>	Assigns the IP address of a BGP neighbor to a peer group.
Step 7	<p>address-family <i>ipv4</i> [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 multicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. This is the default. • The multicast keyword specifies that IPv4 multicast address prefixes will be exchanged. • The vrf keyword and <i>vrf-name</i> argument specify that IPv4 VRF instance information will be exchanged.
Step 8	<p>neighbor <i>peer-group-name</i> activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor fingroup activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.</p> <p>Note By default, neighbors that are defined using the neighbor remote-as command in router configuration mode exchange only unicast address prefixes. To allow BGP to exchange other address prefix types, such as multicast that is configured in this example, neighbors must also be activated using the neighbor activate command.</p>

	Command or Action	Purpose
Step 9	neighbor <i>ip-address</i> peer-group <i>peer-group-name</i> Example: Device(config-router-af)# neighbor 192.168.1.1 peer-group fingroup	Assigns the IP address of a BGP neighbor to a peer group.
Step 10	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring Peer Session Templates

The following tasks create and configure a peer session template:

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.



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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 101	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-session <i>session-template-name</i> Example: Router(config-router)# template peer-session INTERNAL-BGP	Enters session-template configuration mode and creates a peer session template.
Step 5	remote-as <i>autonomous-system-number</i> Example: Router(config-router-stmp)# remote-as 202	(Optional) Configures peering with a remote neighbor in the specified autonomous system. Note Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.
Step 6	timers <i>keepalive-interval hold-time</i> Example: Router(config-router-stmp)# timers 30 300	(Optional) Configures BGP keepalive and hold timers. <ul style="list-style-type: none"> • The hold time must be at least twice the keepalive time.

	Command or Action	Purpose
		Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 7	end Example: Router(config-router)# end	Exits session-template configuration mode and returns to privileged EXEC mode.
Step 8	show ip bgp template peer-session [<i>session-template-name</i>] Example: Router# show ip bgp template peer-session	Displays locally configured peer session templates. <ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the <i>session-template-name</i> argument. This command also supports all standard output modifiers.

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

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 101</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-session <i>session-template-name</i> Example: <pre>Router(config-router)# template peer-session CORE1</pre>	Enter session-template configuration mode and creates a peer session template.
Step 5	description <i>text-string</i> Example: <pre>Router(config-router-stmp)# description CORE-123</pre>	(Optional) Configures a description. <ul style="list-style-type: none"> • The text string can be up to 80 characters. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 6	update-source <i>interface-type interface-number</i> Example: <pre>Router(config-router-stmp)# update-source loopback 1</pre>	(Optional) Configures a router to select a specific source or interface to receive routing table updates. <ul style="list-style-type: none"> • The 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. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 7	inherit peer-session <i>session-template-name</i> Example: <pre>Router(config-router-stmp)# inherit peer-session INTERNAL-BGP</pre>	Configures this peer session template to inherit the configuration of another peer session template. <ul style="list-style-type: none"> • The example configures this peer session template to inherit the configuration from INTERNAL-BGP. This template can be applied to a neighbor, and the configuration INTERNAL-BGP will be applied indirectly. No additional peer session templates can be directly applied. However, the directly inherited template can contain up to seven indirectly inherited peer session templates.

	Command or Action	Purpose
Step 8	end Example: <pre>Router(config-router)# end</pre>	Exits session-template configuration mode and enters privileged EXEC mode.
Step 9	show ip bgp template peer-session <i>[session-template-name]</i> Example: <pre>Router# show ip bgp template peer-session</pre>	Displays locally configured peer session templates. <ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the optional <i>session-template-name</i> argument. This command also supports all standard output modifiers.

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

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 101</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 172.16.0.1 remote-as 202</pre>	<p>Configures a peering session with the specified neighbor.</p> <ul style="list-style-type: none"> 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	<p>neighbor <i>ip-address</i> inherit peer-session <i>session-template-name</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1</pre>	<p>Sends a peer session template to a neighbor so that the neighbor can inherit the configuration.</p> <ul style="list-style-type: none"> 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	<p>end</p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 7	<p>show ip bgp template peer-session [<i>session-template-name</i>]</p> <p>Example:</p> <pre>Router# show ip bgp template peer-session</pre>	<p>Displays locally configured peer session templates.</p> <ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the optional <i>session-template-name</i> 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](#), on page 140.

Configuring Peer Policy Templates

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

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	<p>template peer-policy <i>policy-template-name</i></p> <p>Example:</p> <pre>Device(config-router)# template peer-policy GLOBAL</pre>	Enters policy-template configuration mode and creates a peer policy template.
Step 5	<p>maximum-prefix <i>prefix-limit [threshold]</i> [restart <i>restart-interval</i> warning-only]</p> <p>Example:</p> <pre>Device(config-router-ptmp)# maximum-prefix 10000</pre>	<p>(Optional) Configures the maximum number of prefixes that a neighbor will accept from this peer.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 6	<p>weight <i>weight-value</i></p> <p>Example:</p> <pre>Device(config-router-ptmp)# weight 300</pre>	<p>(Optional) Sets the default weight for routes that are sent from this neighbor.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 7	<p>prefix-list <i>prefix-list-name {in out}</i></p> <p>Example:</p> <pre>Device(config-router-ptmp)# prefix-list NO-MARKETING in</pre>	<p>(Optional) Filters prefixes that are received by the router or sent from the router.</p> <ul style="list-style-type: none"> The prefix list in the example filters inbound internal addresses. <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router-ptmp)# end</pre>	Exits policy-template configuration mode and returns to privileged EXEC mode.

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 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 configures 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 Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, 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

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	<p>template peer-policy <i>policy-template-name</i></p> <p>Example:</p> <pre>Router(config-router)# template peer-policy NETWORK1</pre>	Enter policy-template configuration mode and creates a peer policy template.
Step 5	<p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-ptmp)# route-map ROUTE in</pre>	<p>(Optional) Applies the specified route map to inbound or outbound routes.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the Peer Policy Templates, on page 84.</p>
Step 6	<p>inherit peer-policy <i>policy-template-name</i> <i>sequence-number</i></p> <p>Example:</p> <pre>Router(config-router-ptmp)# inherit peer-policy GLOBAL 10</pre>	<p>Configures the peer policy template to inherit the configuration of another peer policy template.</p> <ul style="list-style-type: none"> • The <i>sequence-number</i> 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. • 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. • This template in the example will be evaluated first if no other templates are configured with a lower sequence number.
Step 7	<p>end</p> <p>Example:</p> <pre>Router(config-router-ptmp)# end</pre>	Exits policy-template configuration mode and returns to privileged EXEC mode.
Step 8	<p>show ip bgp template peer-policy [<i>policy-template-name</i>[detail]]</p>	Displays locally configured peer policy templates.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router# show ip bgp template peer-policy NETWORK1 detail</pre>	<ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the <i>policy-template-name</i> argument. This command also supports all standard output modifiers. Use the detail keyword to display detailed policy information. <p>Note The detail keyword is supported only in Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases.</p>

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 policies: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 Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, 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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.1.2 remote-as 40000	Configures a peering session with the specified neighbor. • 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.
Step 5	address-family ipv4 [multicast unicast vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4 unicast	Enters address family configuration mode to configure a neighbor to accept address family-specific command configurations.
Step 6	neighbor <i>ip-address</i> inherit peer-policy <i>policy-template-name</i>	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL</pre>	<ul style="list-style-type: none"> 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.
Step 7	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	<p>show ip bgp neighbors [ip-address][policy [detail]]</p> <p>Example:</p> <pre>Router# show ip bgp neighbors 192.168.1.2 policy</pre>	<p>Displays locally configured peer policy templates.</p> <ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the <i>policy-template-name</i> 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. The policy and detail keywords are supported only in Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases. <p>Note Only the syntax required for this task is shown. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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 polices:
  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 information about using BGP peer templates, see the “Configuring Peer Session Templates” and “Configuring Peer Policy Templates” sections.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear ip bgp update-group [<i>index-group</i> <i>ip-address</i>] Example: Device# clear ip bgp update-group 192.168.2.2	Clears BGP update group membership and recalculate BGP update groups. <ul style="list-style-type: none"> • In the example provided, the membership of neighbor 192.168.2.2 is cleared from an update group.
Step 3	show ip bgp replication [<i>index-group</i> <i>ip-address</i>] Example: Device# show ip bgp replication	Displays update replication statistics for BGP update groups.
Step 4	show ip bgp update-group [<i>index-group</i> <i>ip-address</i>] [summary] Example: Device# show ip bgp update-group	Displays information about BGP update groups.

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 you are troubleshooting a problem.

Configuration Examples for a Basic BGP Network

Example: Configuring a BGP Process and Customizing Peers

The following example shows the configuration for Router B in the 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
```

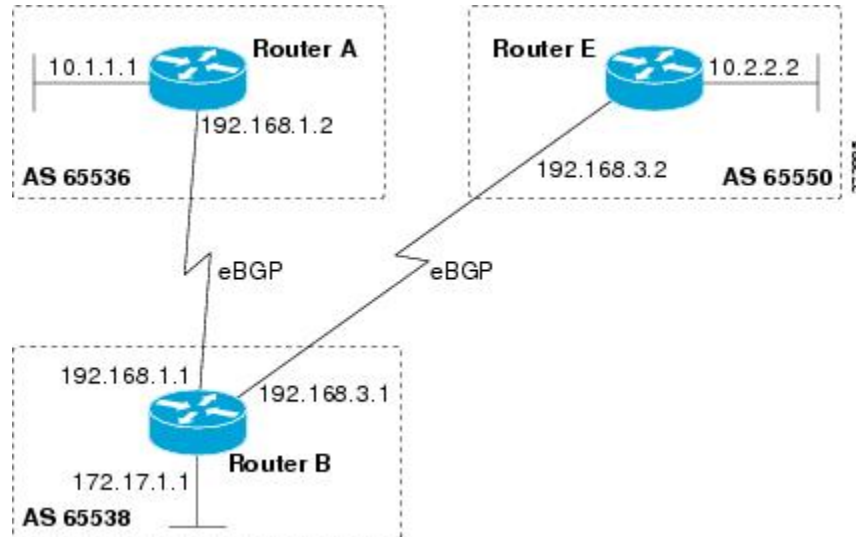
Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Format

The following example shows the configuration for Router A, Router B, and Router E in the figure below with a Border Gateway Protocol (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 14: BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format



Router A

```
router bgp 65536
  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 65538
  !
  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 65538
  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 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

```

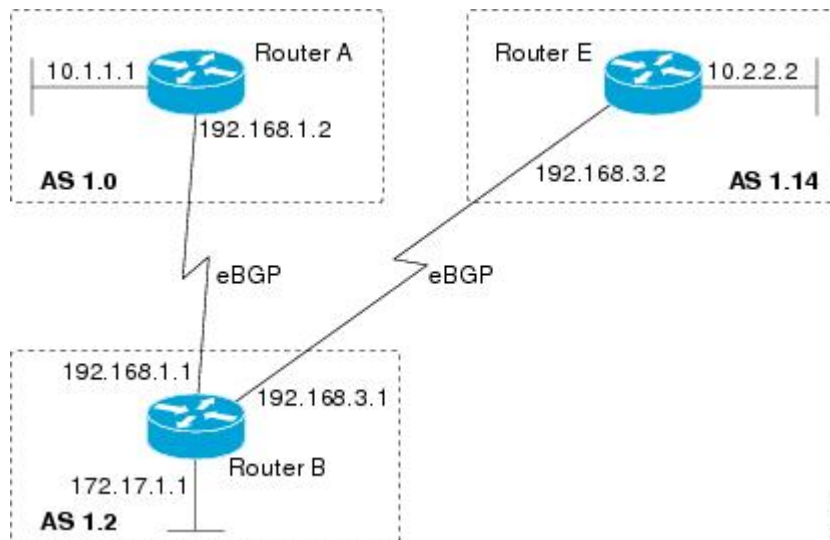
router bgp 65550
  bgp router-id 10.2.2.99
  no bgp default ipv4-unicast
  bgp fast-external-fallover
  bgp log-neighbor-changes
  timers bgp 70 120
  neighbor 192.168.3.1 remote-as 65538
  !
  address-family ipv4
    neighbor 192.168.3.1 activate
    no auto-summary
    no synchronization
    network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

```

Asdot Format

The following example 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 15: BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format

**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
```

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

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SX11, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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
```

4-Byte Autonomous System Number RD Support

The following example shows how to create a VRF with a route distinguisher that contains a 4-byte AS number 65536, and a route target that contains a 4-byte autonomous system number, 65537:

```
ip vrf vpn_red
 rd 65536:100
 route-target both 65537:100
 exit
```

After the configuration is completed, use the **show vrf** command to verify that the 4-byte AS number route distinguisher is set to 65536:100:

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
 rd 65536:100
!
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T 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 the extended community value 1.1:100 for routes that are permitted by the route map.



Note

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SX11, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
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
```

Asdot Default Format for 4-Byte Autonomous System Number RD Support

The following example works if you have configured asdot as the default display format using the **bgp asnotation dot** command:

```
ip vrf vpn_red
 rd 1.0:100
 route-target both 1.1:100
 exit
```

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:

```
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 permit 10
 match ip address prefix-list MULTICAST-PREFIXES
 set nlri multicast
!
route-map MULTICAST-PREFIXES permit 10
!
route-map REDISTRIBUTE-UNICAST permit 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 permit 10
    match ip address prefix-list MULTICAST-PREFIXES
  !
  route-map REDISTRIBUTE-MULTICAST permit 10
    match ip address prefix-list MULTICAST-PREFIXES
  !
  route-map MULTICAST-PREFIXES permit 10
  !
  route-map REDISTRIBUTE-UNICAST permit 20
    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 first the CLI configuration to enable the redistribution of BGP routes into EIGRP using a route map and then 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 192.168.1.2 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.

Example: Dynamic Inbound Soft Reset

The following example shows the 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
```

Example: Inbound Soft Reset Using Stored Information

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
```

Example: 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. 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
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
*> 172.17.1.0/24  0.0.0.0             0             32768  i
```

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.

```
RouterB#
*Nov 30 23:25:55.995: %BGP-5-ADJCHANGE: neighbor 192.168.3.2 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. In Cisco IOS Release 12.2(25)S and later releases, the syntax is **clear ip bgp all**. 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
0x2FB5C90   1      4      0 i
0x2FB5C00  1361    2      0 50000 i
0x2FB5D20  2625    2      0 40000 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	0	00:03:49	1
192.168.3.2	4	50000	468	467	0	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 configures BGP to not advertise inactive routes:

```
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

The following example configures a maximum route limit in the VRF named RED and configures BGP to not advertise inactive routes through the VRF named RED:

```
Device(config)# ip vrf RED
Device(config-vrf)# rd 50000:10
Device(config-vrf)# maximum routes 1000 10
Device(config-vrf)# exit
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 vrf RED
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```


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:

```
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:

```
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.

```
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
```

Examples: Configuring Peer Policy Templates

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

```
router bgp 45000
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.

```
router bgp 45000
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 named CUSTOMER-A. Assuming this example is a continuation of the example above, because the peer policy template named CUSTOMER-A above inherited the configuration from the templates named PRIMARY-IN and GLOBAL, 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-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 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 10.4.9.8 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

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	“Cisco BGP Overview” module in the <i>IP Routing: BGP Configuration Guide</i>
Multiprotocol Label Switching (MPLS) and BGP configuration example using the IPv4 VRF address family	“MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels” module in the <i>MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide</i>

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	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

RFCs

RFC	Title
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring a Basic BGP Network

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 [http://www.cisco.com/cisco/web/featurenavigator/index.html](#). An account on Cisco.com is not required.

Table 13: Feature Information for Configuring a Basic BGP Network

Feature Name	Releases	Feature Configuration Information
BGP Conditional Route Injection	12.0(22)S 12.2(4)T 12.2(14)S 15.0(1)S Cisco IOS XE 3.1.0SG	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.
BGP Configuration Using Peer Templates	12.0(24)S 12.2(18)S 12.2(27)SBC 12.3(4)T 15.0(1)S	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.

Feature Name	Releases	Feature Configuration Information
BGP Dynamic Update Peer Groups	12.0(24)S 12.2(18)S 12.2(27)SBC 12.3(4)T 15.0(1)S Cisco IOS XE 3.1.0SG	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 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.
BGP Hybrid CLI	12.0(22)S 12.2(15)T 15.0(1)S	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.
Suppress BGP Advertisement for Inactive Routes	12.2(25)S 12.2(33)SXH 15.0(1)M 15.0(1)S	The Suppress BGP Advertisements for Inactive Routes feature allows you to configure the suppression of advertisements for routes that are not installed in the Routing Information Base (RIB). Configuring this feature allows Border Gateway Protocol (BGP) updates to be more consistent with data used for traffic forwarding.



BGP Support for 4-byte ASN

The Cisco implementation of 4-byte autonomous system (AS) numbers uses `asplain` (65538, for example) as the default regular expression match and the output display format for AS numbers. However, you can configure 4-byte AS numbers in both the `asplain` format and the `asdot` format as described in RFC 5396. In addition, 4-byte ASN route distinguisher (RD) and route target (RT) BGP support for 4-byte autonomous numbers is added.

- [Finding Feature Information, page 167](#)
- [Information About BGP Support for 4-byte ASN, page 167](#)
- [How to Configure BGP Support for 4-byte ASN, page 171](#)
- [Configuration Examples for BGP Support for 4-byte ASN, page 177](#)
- [Additional References for BGP Support for 4-byte ASN, page 182](#)
- [Feature Information for BGP Support for 4-byte ASN, page 183](#)

Finding Feature Information

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

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

Information About BGP Support for 4-byte ASN

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system (AS) 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 AS numbers, the Internet Assigned Number Authority (IANA) started to allocate

four-octet AS numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, documents three methods of representing AS numbers. Cisco has implemented the following two methods:

- **Asplain**—Decimal value notation where both 2-byte and 4-byte AS numbers are represented by their decimal value. For example, 65526 is a 2-byte AS number and 234567 is a 4-byte AS number.
- **Asdot**—Autonomous system dot notation where 2-byte AS numbers are represented by their decimal value and 4-byte AS numbers are represented by a dot notation. For example, 65526 is a 2-byte AS number and 1.169031 is a 4-byte AS 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) AS numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte AS 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 AS numbers are configured, matched in regular expressions, and displayed in **show** command output in Cisco IOS images where only asdot formatting is available.

Table 14: Asdot Only 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default AS Number Formatting

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte AS numbers uses asplain as the default display format for AS numbers, but you can configure 4-byte AS numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte AS numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte AS 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 AS 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 AS numbers in either asplain or asdot format, only one format is used to display **show** command output and control 4-byte AS number matching for regular expressions, and the default is asplain format. To display 4-byte AS 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 AS numbers, you can still use 2-byte AS numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte AS numbers regardless of the format configured for 4-byte AS numbers.

Table 15: Default Asplain 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 16: Asdot 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private AS 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 AS numbers to 4-byte AS numbers. A new reserved (private) AS number, 23456, was created by RFC 4893 and this number cannot be configured as an AS number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved AS 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 AS number registry. Reserved 2-byte AS numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte AS numbers are from 65536 to 65551 inclusive.

Private 2-byte AS numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private AS 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 AS numbers to external networks. Cisco IOS software does not remove private AS numbers from routing updates by default. We recommend that ISPs filter private AS numbers.



Note

AS number assignment for public and private networks is governed by the IANA. For information about AS numbers, including reserved number assignment, or to apply to register an AS number, see the following URL: <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 (AS) numbers uses asplain—65538, for example—as the default regular expression match and output display format for AS numbers, but you can configure 4-byte AS 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 AS 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 AS number formats, see the “BGP Autonomous System Number Formats” section.

In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte AS 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” section.

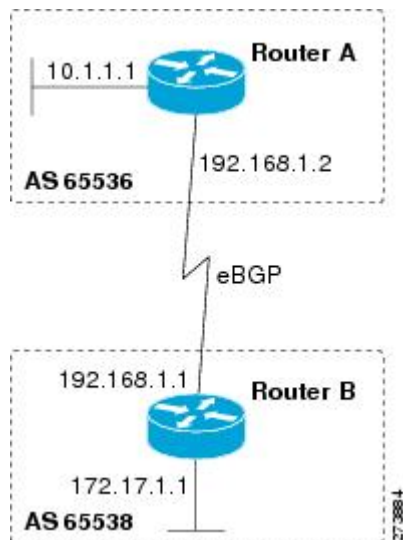
Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte AS numbers to 4-byte AS numbers. To ensure a smooth transition, we recommend that all BGP speakers within an AS that is identified using a 4-byte AS number be upgraded to support 4-byte AS numbers.



Note

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

Figure 16: BGP Peers in Two Autonomous Systems Using 4-Byte Numbers



How to Configure BGP Support for 4-byte ASN

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

Perform this task to configure a Border Gateway Protocol (BGP) routing process and BGP peers when the BGP peers are located in an autonomous system (AS) that uses 4-byte AS 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 (in the “Cisco Implementation of 4-Byte Autonomous System Numbers” section). The 4-byte AS numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in AS number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before You Begin



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

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 65538</pre>	<p>Enters router configuration mode for the specified routing process.</p> <ul style="list-style-type: none"> In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 65536</pre>	<p>Adds the IP address of the neighbor in the specified AS to the IPv4 multiprotocol BGP neighbor table of the local device.</p> <ul style="list-style-type: none"> In this example, the 4-byte AS number, 65536, is defined in asplain notation.
Step 5	Repeat Step 4 to define other BGP neighbors, as required.	--
Step 6	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> The unicast keyword specifies the IPv4 unicast address family. By default, the device 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 <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.2 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.

	Command or Action	Purpose
Step 8	Repeat Step 7 to activate other BGP neighbors, as required.	--
Step 9	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] Example: Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0	(Optional) Specifies a network as local to this AS and adds it to the BGP routing table. <ul style="list-style-type: none"> For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 10	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	show ip bgp [<i>network</i>] [<i>network-mask</i>] Example: Device# show ip bgp 10.1.1.0	(Optional) Displays the entries in the BGP routing table. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 12	show ip bgp summary Example: Device# show ip bgp summary	(Optional) Displays the status of all BGP connections.

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 AS 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
    192.168.1.2 from 192.168.1.2 (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 AS 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:

```
RouterB# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 65538
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
```

```

3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 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 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  Stated
192.168.1.2   4      65536    6      6       3    0    0 00:01:33    1

```

Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP devices.

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 (AS) 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 AS numbers.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	show ip bgp summary Example: Device# show ip bgp summary	Displays the status of all Border Gateway Protocol (BGP) connections.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65538	Enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 5	bgp asnotation dot Example: Device(config-router)# bgp asnotation dot	Changes the default output format of BGP 4-byte AS numbers from asplain (decimal values) to dot notation. <p>Note 4-byte AS 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.</p>
Step 6	end Example: Device(config-router)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	clear ip bgp * Example: Device# clear ip bgp *	Clears and resets all current BGP sessions. <ul style="list-style-type: none"> In this example, a hard reset is performed to ensure that the 4-byte AS number format change is reflected in all BGP sessions. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 8	show ip bgp summary Example: Device# show ip bgp summary	Displays the status of all BGP connections.
Step 9	show ip bgp regexp <i>regexp</i> Example: Device# show ip bgp regexp ^1\.0\$	Displays routes that match the AS path regular expression. <ul style="list-style-type: none"> In this example, a regular expression to match a 4-byte AS path is configured using asdot format.

	Command or Action	Purpose
Step 10	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 11	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65538	Enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 12	no bgp asnotation dot Example: Device(config-router)# no bgp asnotation dot	Resets the default output format of BGP 4-byte AS numbers back to asplain (decimal values). Note 4-byte AS 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: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.
Step 14	clear ip bgp * Example: Device# clear ip bgp *	Clears and resets all current BGP sessions. <ul style="list-style-type: none"> • In this example, a hard reset is performed to ensure that the 4-byte AS 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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .

Examples

The following output from the **show ip bgp summary** command shows the default asplain format of the 4-byte AS numbers. Note the asplain format of the 4-byte AS 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    0 00:03:04    0
192.168.3.2   4      65550     4       4        1    0    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 AS numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 AS 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
192.168.1.2   4        1.0      9      9        1    0    0 00:04:13    0
192.168.3.2   4        1.14     6      6        1    0    0 00:01:24    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 regular expression match format for 4-byte AS paths is changed to asdot notation format. Although a 4-byte AS number can be configured in a regular expression using either asplain format or asdot format, only 4-byte AS 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 AS 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 AS 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.

```
Router# show ip bgp regexp ^65536$

Router# show ip bgp regexp ^1\.0$

BGP table version is 2, 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 1.0 i
```

Configuration Examples for BGP Support for 4-byte ASN

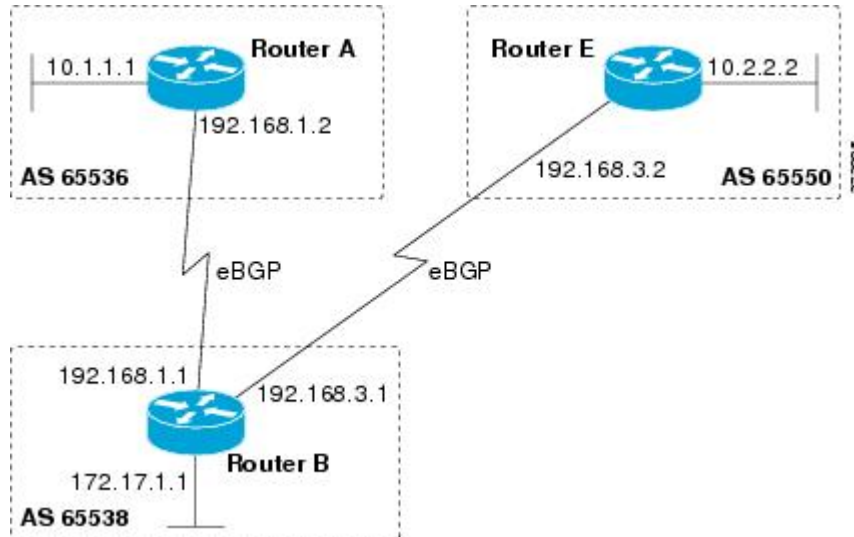
Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Format

The following example shows the configuration for Router A, Router B, and Router E in the figure below with a Border Gateway Protocol (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 17: BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format



Router A

```
router bgp 65536
  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 65538
  !
  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 65538
  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 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

```

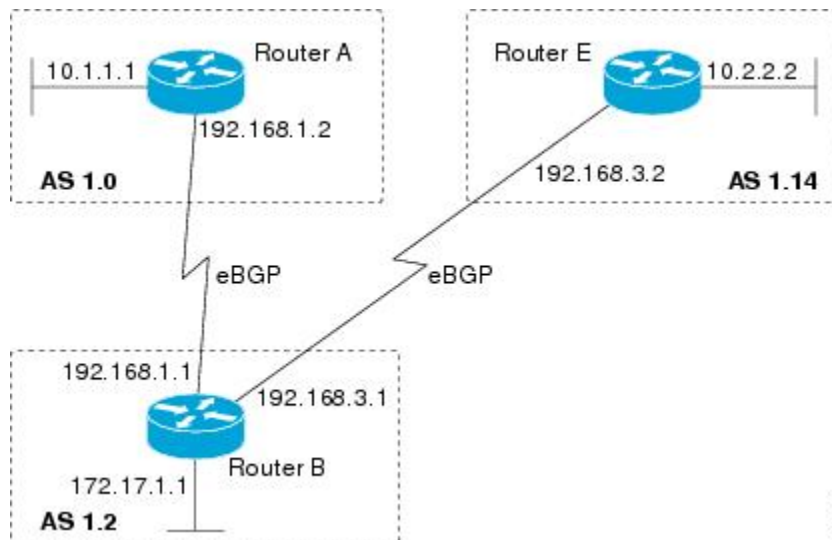
router bgp 65550
  bgp router-id 10.2.2.99
  no bgp default ipv4-unicast
  bgp fast-external-fallover
  bgp log-neighbor-changes
  timers bgp 70 120
  neighbor 192.168.3.1 remote-as 65538
  !
  address-family ipv4
    neighbor 192.168.3.1 activate
    no auto-summary
    no synchronization
    network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

```

Asdot Format

The following example 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 18: BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format

**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
```

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

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SX11, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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
```

4-Byte Autonomous System Number RD Support

The following example shows how to create a VRF with a route distinguisher that contains a 4-byte AS number 65536, and a route target that contains a 4-byte autonomous system number, 65537:

```
ip vrf vpn_red
 rd 65536:100
 route-target both 65537:100
 exit
```

After the configuration is completed, use the **show vrf** command to verify that the 4-byte AS number route distinguisher is set to 65536:100:

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
 rd 65536:100
!
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T 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 the extended community value 1.1:100 for routes that are permitted by the route map.



Note

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SX11, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
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
```

Asdot Default Format for 4-Byte Autonomous System Number RD Support

The following example works if you have configured asdot as the default display format using the **bgp asnotation dot** command:

```
ip vrf vpn_red
 rd 1.0:100
 route-target both 1.1:100
 exit
```

Additional References for BGP Support for 4-byte ASN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5396	<i>Textual Representation of Autonomous System (AS) Numbers</i>
RFC 5398	<i>Autonomous System (AS) Number Reservation for Documentation Use</i>
RFC 5668	<i>4-Octet AS Specific BGP Extended Community</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for 4-byte ASN

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 17: Feature Information for BGP Support for 4-byte ASN

Feature Name	Releases	Feature Information
BGP Support for 4-byte ASN	12.0(32)SY8 12.2(33)SX11	The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. The following commands were introduced or modified: 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 extcommunity , set origin , all show ip bgp commands that display an autonomous system number, and show ip extcommunity-list .
BGP—4-Byte ASN RD and RT Support		The BGP Support for 4-Byte ASN RD and RT support for 4-byte autonomous system numbers was added.



IPv6 Routing: Multiprotocol BGP Extensions for IPv6

- [Finding Feature Information, page 185](#)
- [Information About IPv6 Routing: Multiprotocol BGP Extensions for IPv6, page 185](#)
- [How to Implement Multiprotocol BGP for IPv6, page 186](#)
- [Configuration Examples for Multiprotocol BGP for IPv6, page 197](#)
- [Additional References, page 198](#)
- [Feature Information for IPv6 Routing: Multiprotocol BGP Extensions for IPv6, page 199](#)

Finding Feature Information

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

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

Information About IPv6 Routing: Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP is the supported Exterior Gateway Protocol (EGP) for IPv6. Multiprotocol BGP extensions for IPv6 supports many of the same features and functionality as IPv4 BGP. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and Network Layer Reachability Information (NLRI) and next hop (the next device in the path to the destination) attributes that use IPv6 addresses.

How to Implement Multiprotocol BGP for IPv6

Configuring an IPv6 BGP Routing Process and BGP Router ID

Perform this task to configure an IPv6 BGP routing process and an optional BGP router ID for a BGP-speaking device.

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

When configuring BGP on a device that is enabled only for IPv6 (that is, the device does not have an IPv4 address), you must manually configure the BGP router ID for the device. The BGP router ID, which is represented as a 32-bit value using an IPv4 address syntax, must be unique to the BGP peers of the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **bgp router-id** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process, and enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as command unless you configure the no bgp default ipv4-unicast command before configuring the neighbor remote-as command.
Step 5	bgp router-id ip-address Example: Device(config-router)# bgp router-id 192.168.99.70	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.

Configuring IPv6 Multiprotocol BGP Between Two Peers

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, as shown for IPv6 prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp as-number**
4. **neighbor** {*ip-address* | *ipv6-address [%]* | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
5. **address-family ipv6** [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address %*} **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp as-number Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor {ip-address ipv6-address [%] peer-group-name} remote-as autonomous-system-number [alternate-as autonomous-system-number ...] Example: Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	address-family ipv6 [unicast multicast] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor {ip-address peer-group-name ipv6-address %} activate Example: Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.

Advertising Routes into IPv6 Multiprotocol BGP

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Device(config-router)# address-family ipv6 unicast	Specifies the IPv6 address family, and enters address family configuration mode. • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Device(config-router-af)# network 2001:DB8::/24	Advertises (injects) the specified prefix into the IPv6 BGP database (the routes must first be found in the IPv6 unicast routing table). • The prefix is injected into the database for the address family specified in the previous step. • Routes are tagged from the specified prefix as “local origin.” • The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 6	exit Example: Device(config-router-af)# exit	Exits address family configuration mode, and returns the device to router configuration mode. <ul style="list-style-type: none"> Repeat this step to exit router configuration mode and return the device to global configuration mode.

Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

- 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, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *as-number*
- neighbor** {*ip-address* | *ipv6-address*[%] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
- address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
- neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* %} **activate**
- neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [%]} **route-map** *map-name* {**in** | **out**}
- exit**
- exit**
- route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
- match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> [%] <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number</i> ...] Example: Device(config-router)# neighbor 2001:DB8:0:cc00::1 remote-as 64600	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> %} activate Example: Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device using the specified link-local addresses.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} route-map <i>map-name</i> { in out }	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> • Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 route-map rtp in</pre>	Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode, and returns the device to router configuration mode.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode, and returns the device to global configuration mode.
Step 10	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map rtp permit 10</pre>	Defines a route map and enters route-map configuration mode. <ul style="list-style-type: none"> Follow this step with a match command.
Step 11	<p>match ipv6 address {prefix-list <i>prefix-list-name</i> <i>access-list-name</i>}</p> <p>Example:</p> <pre>Device(config-route-map)# match ipv6 address prefix-list cisco</pre>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.

Redistributing Prefixes into IPv6 Multiprotocol BGP

Redistribution is the process of redistributing, or injecting, prefixes from one routing protocol into another routing protocol. This task explains how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*] [*source-protocol-options*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	redistribute bgp [<i>process-id</i>] [metric <i>metric-value</i>] [route-map <i>map-name</i>] [<i>source-protocol-options</i>] Example: Device(config-router-af)# redistribute bgp 64500 metric 5	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 6	exit Example: Device(config-router-af)# exit	Exits address family configuration mode, and returns the device to router configuration mode. <ul style="list-style-type: none"> • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Clearing External BGP Peers

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} external [soft] [in | out]`
3. `clear bgp ipv6 {unicast | multicast} peer-group name`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>clear bgp ipv6 {unicast multicast} external [soft] [in out]</code></p> <p>Example:</p> <pre>Device# clear bgp ipv6 unicast external soft in</pre>	<p>Clears external IPv6 BGP peers.</p>
Step 3	<p><code>clear bgp ipv6 {unicast multicast} peer-group <i>name</i></code></p> <p>Example:</p> <pre>Device# clear bgp ipv6 unicast peer-group marketing</pre>	<p>Clears all members of an IPv6 BGP peer group.</p>

Advertising IPv4 Routes Between IPv6 BGP Peers

If an IPv6 network is connecting two separate IPv4 networks, IPv6 can be used to advertise the IPv4 routes. Configure the peering using the IPv6 addresses within the IPv4 address family. Set the next hop with a static route or with an inbound route map because the advertised next hop will usually be unreachable. Advertising IPv6 routes between two IPv4 peers is also possible using the same model.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address*[%] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [%]} **route-map** *map-name* {**in** | **out**}
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address* [... *ip-address*] [**peer-address**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>peer-group-name</i> peer-group Example: Device(config-router)# neighbor 6peers peer-group	Creates a multiprotocol BGP peer group.

	Command or Action	Purpose
Step 5	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i>[%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number</i> ...]</p> <p>Example:</p> <pre>Device(config-router)# neighbor 6peers remote-as 65002</pre>	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 6	<p>address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	<p>neighbor <i>ipv6-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:1234::2 peer-group 6peers</pre>	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 6peers route-map rmap out</pre>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode, and returns the device to router configuration mode.
Step 10	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode, and returns the device to global configuration mode.
Step 11	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map rmap permit 10</pre>	Defines a route map and enters route-map configuration mode.
Step 12	<p>set ip next-hop <i>ip-address</i> [... <i>ip-address</i>] [peer-address]</p> <p>Example:</p> <pre>Device(config-route-map)# set ip next-hop 10.21.8.10</pre>	Overrides the next hop advertised to the peer for IPv4 packets.

Configuration Examples for Multiprotocol BGP for IPv6

Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer

The following example enables IPv6 globally, configures a BGP process, and establishes a BGP router ID. Also, the IPv6 multiprotocol BGP peer 2001:DB8:0:CC00::1 is configured and activated.

```
ipv6 unicast-routing
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp router-id 192.168.99.70
 neighbor 2001:DB8:0:CC00::1 remote-as 64600
 address-family ipv6 unicast
  neighbor 2001:DB8:0:CC00::1 activate
```

Example: Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
 no bgp default ipv4-unicast
 neighbor group1 peer-group
 neighbor 2001:DB8:0:CC00::1 remote-as 64600
 address-family ipv6 unicast
  neighbor group1 activate
  neighbor 2001:DB8:0:CC00::1 peer-group group1
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
router bgp 65000
 no bgp default ipv4-unicast
 address-family ipv6 unicast
  network 2001:DB8::/24
```

Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

The following example configures the route map named rtp to permit IPv6 unicast routes from network 2001:DB8::/24 if they match the prefix list named cisco:

```
router bgp 64900
 no bgp default ipv4-unicast
```

```

neighbor 2001:DB8:0:CC00::1 remote-as 64700
address-family ipv6 unicast
neighbor 2001:DB8:0:CC00::1 activate
neighbor 2001:DB8:0:CC00::1 route-map rtp in
ipv6 prefix-list cisco seq 10 permit 2001:DB8::/24
route-map rtp permit 10
match ipv6 address prefix-list cisco

```

Example: Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes RIP routes into the IPv6 unicast database of the local device:

```

router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 unicast
redistribute rip

```

Example: Advertising IPv4 Routes Between IPv6 Peers

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. Peering is configured using IPv6 addresses in the IPv4 address family configuration mode. The inbound route map named rmap sets the next hop because the advertised next hop is likely to be unreachable.

```

router bgp 65000
!
neighbor 6peers peer-group
neighbor 2001:DB8:1234::2 remote-as 65002
address-family ipv4
neighbor 6peers activate
neighbor 6peers soft-reconfiguration inbound
neighbor 2001:DB8:1234::2 peer-group 6peers
neighbor 2001:DB8:1234::2 route-map rmap in
!
route-map rmap permit 10
set ip next-hop 10.21.8.10

```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: Multiprotocol BGP Extensions for IPv6

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 18: Feature Information for IPv6 Routing: Multiprotocol BGP Extensions for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: Multiprotocol BGP Extensions for IPv6	12.2(17a)SX1	Multiprotocol BGP Extensions for IPv6 supports the same features and functionality as IPv4 BGP.



IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

- [Finding Feature Information, page 201](#)
- [Information About IPv6 Routing: Multiprotocol BGP Link-Local Address Peering, page 201](#)
- [How to Configure IPv6 Routing: Multiprotocol BGP Link-Local Address Peering, page 202](#)
- [Configuration Examples for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering, page 206](#)
- [Additional References, page 207](#)
- [Feature Information for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering, page 208](#)

Finding Feature Information

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

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

Information About IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

IPv6 Multiprotocol BGP Peering Using a Link-Local Address

The IPv6 multiprotocol BGP can be configured between two IPv6 devices (peers) using link-local addresses. For this function to work, you must identify the interface for the neighbor by using the **neighbor update-source** command, and you must configure a route map to set an IPv6 global next hop.

Border Gateway Protocol (BGP) uses third-party next hops for peering with multiple peers over IPv6 link-local addresses on the same interface. Peering over link-local addresses on different interfaces cannot use third party next hops. The neighbors peering using link-local addresses are split into one update group per interface. BGP splits update group membership for neighbors with link-local addresses based on the interface used to communicate with that neighbor.

How to Configure IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

Configuring IPv6 multiprotocol BGP between two IPv6 devices (peers) using link-local addresses requires that you identify the interface for the neighbor by using the **neighbor update-source** command and that you configure a route map to set an IPv6 global next hop.



Note

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To be able 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, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.
- The route-map used to modify the next hop needs to be applied outbound only. Inbound route-map to modify next-hop ipv6 address is not supported. Inbound route-map is supported only for IPv4 address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* [%]} **peer-group**}
5. **neighbor** {*ip-address* | *ipv6-address* [%] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
6. **neighbor** {*ip-address* | *ipv6-address* [%] | *peer-group-name*} **update-source** *interface-type* *interface-number*
7. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpn**]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* %} **activate**
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [%]} **route-map** *map-name* {**in** | **out**}
10. **exit**
11. **exit**
12. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
13. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
14. **set ipv6 next-hop** *ipv6-address* [*link-local-address*] [**peer-address**]
15. **exit**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> [%]} peer-group }	Configures a BGP neighbor to member of a peer group. Note % keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% peer-group</pre>	
Step 5	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> [%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number</i> ...]</p> <p>Example:</p> <pre>Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% remote-as 64600</pre>	<p>Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.</p> <p>Note % keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.</p>
Step 6	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> [%] <i>peer-group-name</i>} update-source <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% update-source GigabitEthernet 0/0</pre>	<p>Specifies the link-local address over which the peering is to occur.</p> <ul style="list-style-type: none"> The optional % keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface. If there are multiple connections to the neighbor and you do not specify the neighbor interface by using the <i>interface-type</i> and <i>interface-number</i> arguments in the neighbor update-source command, a TCP connection cannot be established with the neighbor using link-local addresses.
Step 7	<p>address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv6</pre>	<p>Specifies the IPv6 address family, and enters address family configuration mode.</p> <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 8	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv6 address family with the local device using the specified link-local addresses.</p> <ul style="list-style-type: none"> The optional % keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} route-map <i>map-name</i> {in out}</p>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> The optional % keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% route-map nh6 out</pre>	
Step 10	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode, and returns the device to router configuration mode.
Step 11	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode, and returns the device to global configuration mode.
Step 12	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map nh6 permit 10</pre>	Defines a route map and enters route-map configuration mode.
Step 13	<p>match ipv6 address {prefix-list <i>prefix-list-name</i> <i>access-list-name</i>}</p> <p>Example:</p> <pre>Device(config-route-map)# match ipv6 address prefix-list cisco</pre>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.
Step 14	<p>set ipv6 next-hop <i>ipv6-address</i> [<i>link-local-address</i>] [peer-address]</p> <p>Example:</p> <pre>Device(config-route-map)# set ipv6 next-hop 2001:DB8::1</pre>	<p>Overrides the next hop advertised to the peer for IPv6 packets that pass a match clause of a route map for policy routing.</p> <ul style="list-style-type: none"> • The <i>ipv6-address</i> argument specifies the IPv6 global address of the next hop. It need not be an adjacent device. • The <i>link-local-address</i> argument specifies the IPv6 link-local address of the next hop. It must be an adjacent device. If you do not specify this optional argument, the link-local address of the interface specified with the <i>interface-type</i> argument (in the neighbor update-source command in Step 5) is included as the next-hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses. • The route map sets the IPv6 next-hop addresses (global and link-local) in BGP updates. If the route map is not configured, the next-hop address in the BGP updates defaults to the unspecified IPv6 address (::), which is rejected by the peer.

	Command or Action	Purpose
Step 15	exit Example: Device(config-router-map)# exit	Exits router map configuration mode, and returns the device to router configuration mode.
Step 16	end Example: Device(config-router)# end	Exits router configuration mode, and enters privileged EXEC mode.

Configuration Examples for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Example: Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

The following example configures the IPv6 multiprotocol BGP peer FE80::1234:BFF:FE0E:A471 over GigabitEthernet interface 0/0 and sets the route map named nh6 to include the IPv6 next-hop global address of GigabitEthernet interface 0/0 in BGP updates. The IPv6 next-hop link-local address can be set by the nh6 route map (not shown in the following example) or from the interface specified by the **neighbor update-source** command (as shown in this example).

```

Device> enable
Device# configure terminal
Device(config)# router bgp 5
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% peer-group
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% remote-as 64600
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% update-source GigabitEthernet 0/0
Device(config-router)# address-family ipv6
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% activate
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% route-map nh6 out
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# route-map nh6 permit 10
Device(config-router-map)# match ipv6 address prefix-list cisco
Device(config-router-map)# set ipv6 next-hop 2001:DB8:526::1
Device(config-router-map)# exit
Device(config-router)# end
ipv6 prefix-list cisco permit 2001:DB8:2F22::/48 le 128
ipv6 prefix-list cisco deny ::/0

```


**Note**

If you specify only the global IPv6 next-hop address (the *ipv6-address* argument) with the **set ipv6 next-hop** command after specifying the neighbor interface (the *interface-type* argument) with the **neighbor update-source** command, the link-local address of the interface specified with the *interface-type* argument is included as the next hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 19: Feature Information for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Feature Name	Releases	Feature Information
IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	12.2(17a)SX1	IPv6 supports multiprotocol BGP link-local address peering.



IPv6 Multicast Address Family Support for Multiprotocol BGP

- [Finding Feature Information](#), page 209
- [Information About IPv6 Multicast Address Family Support for Multiprotocol BGP](#), page 209
- [How to Implement IPv6 Multicast Address Family Support for Multiprotocol BGP](#), page 210
- [Configuration Examples for IPv6 Multicast Address Family Support for Multiprotocol BGP](#), page 219
- [Additional References](#), page 220
- [Feature Information for IPv6 Multicast Address Family Support for Multiprotocol BGP](#), page 221

Finding Feature Information

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

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

Information About IPv6 Multicast Address Family Support for Multiprotocol BGP

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are usable only for IP unicast, not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

How to Implement IPv6 Multicast Address Family Support for Multiprotocol BGP

Configuring an IPv6 Peer Group to Perform Multicast BGP Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family ipv6** [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified BGP routing process.
Step 4	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Device(config-router)# neighbor group1 peer-group</pre>	Creates a BGP peer group.
Step 5	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600</pre>	<p>Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multicast BGP neighbor table of the local router.</p> <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 6	<p>address-family ipv6 [<i>unicast</i> <i>multicast</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv6 multicast</pre>	<p>Specifies the IPv6 address family, and enters address family configuration mode.</p> <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified in the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i>} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router.</p> <ul style="list-style-type: none"> To avoid extra configuration steps for each neighbor, use the neighbor activate command with the <i>peer-group-name</i> argument as an alternative in this step.
Step 8	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i>} peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 peer-group group1</pre>	Assigns the IPv6 address of a BGP neighbor to a peer group.

Advertising Routes into IPv6 Multiprotocol BGP

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device (config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Device (config-router)# address-family ipv6 unicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.

	Command or Action	Purpose
Step 5	<p>network {<i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i>} [route-map <i>map-tag</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 2001:DB8::/24</pre>	<p>Advertises (injects) the specified prefix into the IPv6 BGP database (the routes must first be found in the IPv6 unicast routing table).</p> <ul style="list-style-type: none"> • The prefix is injected into the database for the address family specified in the previous step. • Routes are tagged from the specified prefix as “local origin.” • The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	<p>Exits address family configuration mode, and returns the device to router configuration mode.</p> <ul style="list-style-type: none"> • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Redistributing Prefixes into IPv6 Multiprotocol BGP

Redistribution is the process of redistributing, or injecting, prefixes from one routing protocol into another routing protocol. This task explains how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*] [*source-protocol-options*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified BGP routing process.
Step 4	<p>address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv6</pre>	<p>Specifies the IPv6 address family, and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	<p>redistribute bgp [<i>process-id</i>] [metric <i>metric-value</i>] [route-map <i>map-name</i>] [<i>source-protocol-options</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# redistribute bgp 64500 metric 5</pre>	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	<p>Exits address family configuration mode, and returns the device to router configuration mode.</p> <ul style="list-style-type: none"> • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Assigning a BGP Administrative Distance



Caution

Changing the administrative distance of BGP internal routes is not recommended. One problem that can occur is the accumulation of routing table inconsistencies, which can break routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **distance bgp** *external-distance internal-distance local-distance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 100	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [unicast multicast] Example: Device(config-router)# address-family ipv6 multicast	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	distance bgp <i>external-distance internal-distance local-distance</i> Example: Device(config-router)# distance bgp 20 20 200	Assigns a BGP administrative distance.

Generating Translate Updates for IPv6 Multicast BGP

The multicast BGP translate-update feature generally is used in a multicast BGP-capable router that peers with a customer site that has only a BGP-capable router; the customer site has not or cannot upgrade its router

to a multicast BGP-capable image. Because the customer site cannot originate multicast BGP advertisements, the router with which it peers will translate the BGP prefixes into multicast BGP prefixes, which are used for multicast-source RPF lookup.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **address-family ipv6 [unicast | multicast]**
5. **neighbor *ipv6-address* translate-update ipv6 multicast [unicast]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 100	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [unicast multicast] Example: Device(config-router)# address-family ipv6 multicast	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	neighbor <i>ipv6-address</i> translate-update ipv6 multicast [unicast] Example: Device(config-router)# neighbor 2001:DB8:7000::2 translate-update ipv6 multicast	Generates multiprotocol IPv6 BGP updates that correspond to unicast IPv6 updates received from a peer.

Resetting IPv6 BGP Sessions

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} {* | autonomous-system-number | ip-address | ipv6-address | peer-group peer-group-name} [soft] [in | out]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>clear bgp ipv6 {unicast multicast} {* autonomous-system-number ip-address ipv6-address peer-group peer-group-name} [soft] [in out]</code></p> <p>Example:</p> <pre>Device# clear bgp ipv6 unicast peer-group marketing soft out</pre>	<p>Resets IPv6 BGP sessions.</p>

Clearing External BGP Peers

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} external [soft] [in | out]`
3. `clear bgp ipv6 {unicast | multicast} peer-group name`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	clear bgp ipv6 {unicast multicast} external [soft] [in out] Example: Device# clear bgp ipv6 unicast external soft in	Clears external IPv6 BGP peers.
Step 3	clear bgp ipv6 {unicast multicast} peer-group name Example: Device# clear bgp ipv6 unicast peer-group marketing	Clears all members of an IPv6 BGP peer group.

Clearing IPv6 BGP Route Dampening Information

SUMMARY STEPS

1. enable
2. clear bgp ipv6 {unicast | multicast} dampening [ipv6-prefix/prefix-length]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} dampening [ipv6-prefix/prefix-length] Example: Device# clear bgp ipv6 unicast dampening 2001:DB8::/64	Clears IPv6 BGP route dampening information and unsuppresses the suppressed routes.

Clearing IPv6 BGP Flap Statistics

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} flap-statistics [ipv6-prefix/prefix-length | regexp regexp | filter-list list]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} flap-statistics [ipv6-prefix/prefix-length regexp regexp filter-list list] Example: Device# clear bgp ipv6 unicast flap-statistics filter-list 3	Clears IPv6 BGP flap statistics.

Configuration Examples for IPv6 Multicast Address Family Support for Multiprotocol BGP

Example: Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 2001:DB8:0:CC00::1 remote-as 64600
address-family ipv6 unicast
neighbor group1 activate
neighbor 2001:DB8:0:CC00::1 peer-group group1
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
router bgp 65000
 no bgp default ipv4-unicast
 address-family ipv6 unicast
  network 2001:DB8::/24
```

Example: Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes RIP routes into the IPv6 unicast database of the local device:

```
router bgp 64900
 no bgp default ipv4-unicast
 address-family ipv6 unicast
  redistribute rip
```

Example: Generating Translate Updates for IPv6 Multicast BGP

The following example shows how to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates:

```
router bgp 64900
 no bgp default ipv4-unicast
 address-family ipv6 multicast
  neighbor 2001:DB8:7000::2 translate-update ipv6 multicast
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast Address Family Support for Multiprotocol 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 20: Feature Information for IPv6 Multicast Address Family Support for Multiprotocol BGP

Feature Name	Releases	Feature Information
IPv6 Multicast Address Family Support for Multiprotocol BGP	12.0(26)S 12.2(25)S 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(4)T 12.4 12.4(2)T 15.0(1)S	This feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP.



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 223](#)
- [Prerequisites for Connecting to a Service Provider Using External BGP, page 224](#)
- [Restrictions for Connecting to a Service Provider Using External BGP, page 224](#)
- [Information About Connecting to a Service Provider Using External BGP, page 224](#)
- [How to Connect to a Service Provider Using External BGP, page 234](#)
- [Configuration Examples for Connecting to a Service Provider Using External BGP, page 291](#)
- [Where to Go Next, page 300](#)
- [Additional References, page 300](#)
- [Feature Information for Connecting to a Service Provider Using External BGP, page 302](#)

Finding Feature Information

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

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

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 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.
- Policy lists are not supported in versions of Cisco IOS software prior to Cisco IOS Release 12.0(22)S and 12.2(15)T. Reloading a router that is running an older version of Cisco IOS software may cause some routing policy configurations to be lost.

Information About Connecting to a Service Provider Using External BGP

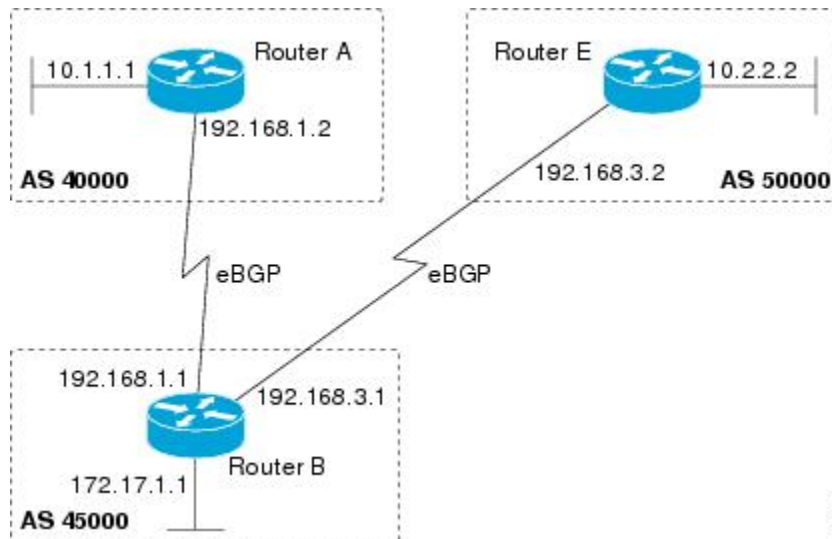
External BGP Peering

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 19: 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 Release 12.0(32)S12, 12.4(24)T, 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 images where only asdot formatting is available.

Table 21: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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 22: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 23: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 12.4(24)T, 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.

**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/>.

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 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 uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, then the oldest paths are selected as multipaths.

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 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

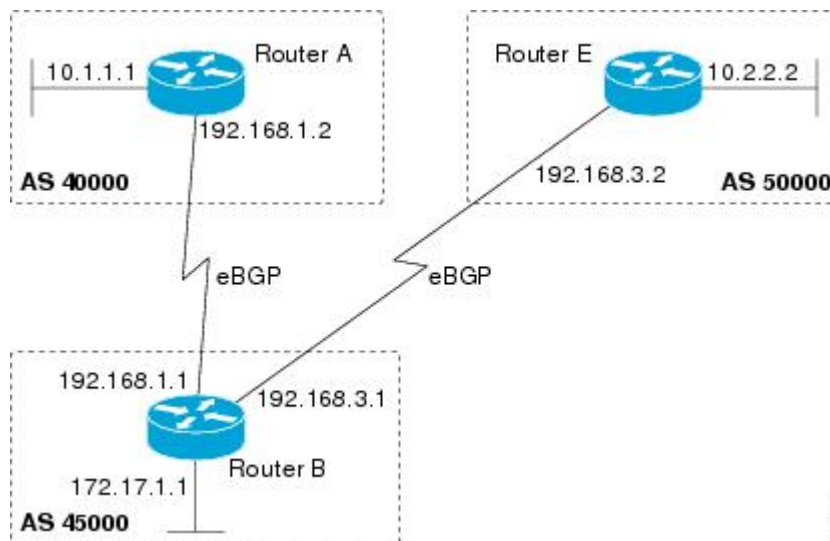
This attribute indicates how the route was included in a BGP routing table. In Cisco software, a route defined using the BGP **network** command is given an origin code of Interior Gateway Protocol (IGP). Routes distributed from an Exterior Gateway Protocol (EGP) are coded with an origin of EGP, and 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 20: 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 24: BGP Policy Processing Order

Inbound	Outbound
Route map	Distribute list
Filter list, AS-path access list, or IP policy	IP prefix list
IP prefix list	Filter list, AS-path access list, or IP policy
Distribute list	Route map



Note

In Cisco IOS Releases 12.0(22)S, 12.2(15)T, 12.2(18)S, 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.

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 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 Attribute

A BGP community is a group of routes that share a common property, regardless of their network, autonomous system, or any physical boundaries. In large networks, applying a common routing policy by using prefix lists or access lists requires individual peer statements on each networking device. Using the BGP COMMUNITIES attribute, BGP speakers with common routing policies can implement inbound or outbound route filters based on the community tag, rather than consult long lists of individual permit or deny statements. A COMMUNITIES attribute can contain multiple communities.

A route can belong to multiple communities. The network administrator defines the communities to which a route belongs. By default, all routes belong to the general Internet community.

In addition to numbered communities, there are several predefined (well-known) communities:

- no-export—Do not advertise this route to external BGP peers.
- no-advertise—Do not advertise this route to any peer.
- internet—Advertise this route to the Internet community. All BGP-speaking networking devices belong to this community.
- local-as—Do not send this route outside the local autonomous system.
- gshut—Community of routes gracefully shut down.

The COMMUNITIES 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 else the COMMUNITIES attribute will be discarded. By default, no COMMUNITIES attribute is sent to a neighbor. In order for a COMMUNITIES attribute to be sent to a neighbor, use the **neighbor send-community** command.

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.

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 lists 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.

**Note**

Policy lists are supported only by BGP and are not supported by other IP routing protocols.

How to Connect to a Service Provider Using External BGP

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
- Multi-Exit Discriminator (MED)

Perform one of the following tasks to influence inbound path selection:

Influencing Inbound Path Selection by Modifying the AS_PATH Attribute

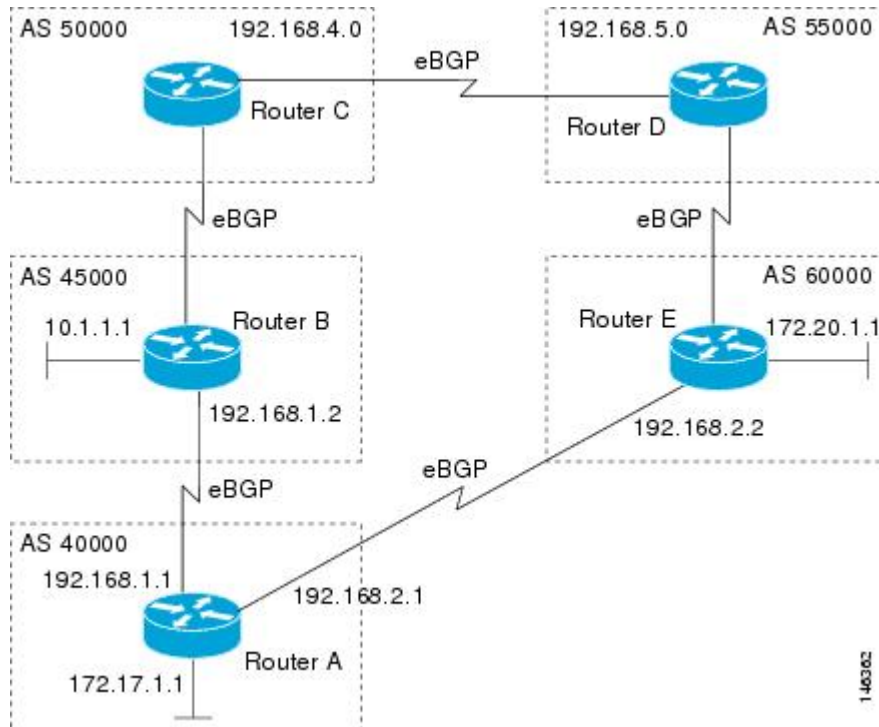
Perform this task to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS_PATH attribute. The configuration is performed at Router A in the figure below. For a configuration example of this task using 4-byte autonomous system numbers in asplain format, see the “Example: Influencing Inbound Path Selection by Modifying the AS_PATH Attribute Using 4-Byte AS Numbers”.

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.

Figure 21: Network Topology for Modifying the AS_PATH Attribute



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*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. **exit-address-family**
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

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 40000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
Step 4	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 45000</pre>	<p>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.</p> <ul style="list-style-type: none"> • 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 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • 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	Purpose
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> } activate Example: <pre>Device(config-router-af)# neighbor 192.168.1.2 activate</pre>	Enables address exchange for address family IPv4 unicast for the BGP neighbor at 192.168.1.2 on Router B.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-map <i>map-name</i> { in out } Example: <pre>Device(config-router-af)# neighbor 192.168.1.2 route-map PREPEND out</pre>	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> In this example, the route map named PREPEND is applied to outbound routes to Router B.
Step 9	exit-address-family Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 10	exit Example: <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 11	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: <pre>Device(config)# route-map PREPEND permit 10</pre>	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named PREPEND is created with a permit clause.
Step 12	set as-path { <i>tag</i> prepend <i>as-path-string</i> } Example: <pre>Device(config-route-map)# set as-path prepend 40000 40000</pre>	Modifies an autonomous system path for BGP routes. <ul style="list-style-type: none"> 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.
Step 13	end Example: <pre>Device(config-route-map)# end</pre>	Exits route map configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 14	show running-config Example: Device# show running-config	Displays the running configuration file.

Examples

The following partial output of the **show running-config** command shows the configuration from this task.

Router A

```
Device# show running-config
.
.
.
router bgp 40000
 neighbor 192.168.1.2 remote-as 45000
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-map PREPEND out
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
 !
 route-map PREPEND permit 10
  set as-path prepend 40000 40000
.
.
.
```

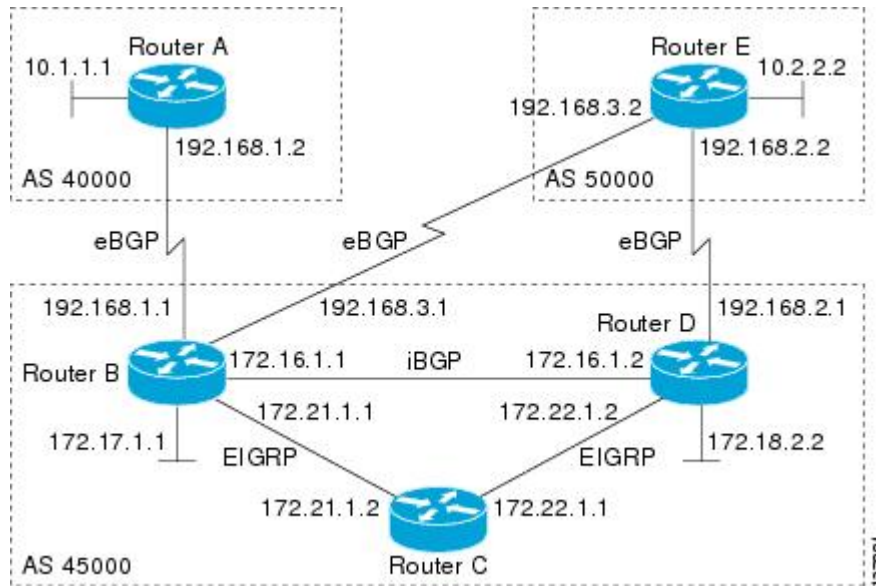
Influencing Inbound Path Selection by Setting the MED Attribute

One of the methods that BGP can use to influence the choice of paths into another autonomous system is to set the Multi-Exit Discriminator (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 22: 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

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
Step 4	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	<p>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.</p>
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.16.1.0 mask 255.255.255.0</pre>	<p>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • 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	Purpose
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-map <i>map-name</i> { in out } Example: <pre>Device(config-router-af)# neighbor 192.168.3.2 route-map MED out</pre>	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> In this example, the route map named MED is applied to outbound routes to the BGP peer at Router E.
Step 8	exit Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 9	exit Example: <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 10	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: <pre>Device(config)# route-map MED permit 10</pre>	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named MED is created.
Step 11	set metric <i>value</i> Example: <pre>Device(config-route-map)# set metric 50</pre>	Sets the MED metric value.
Step 12	end Example: <pre>Device(config-route-map)# end</pre>	Exits route map configuration mode and enters privileged EXEC mode.
Step 13	Repeat Step 1 through Step 12 at Router D.	—
Step 14	show ip bgp [<i>network</i>] [<i>network-mask</i>] Example: <pre>Device# show ip bgp 172.17.1.0 255.255.255.0</pre>	(Optional) Displays the entries in the BGP routing table. <ul style="list-style-type: none"> 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 <i>Cisco IOS IP Routing: BGP Command Reference</i>.

Examples

The following output is from Router E in the figure above 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.

```
Device# 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:

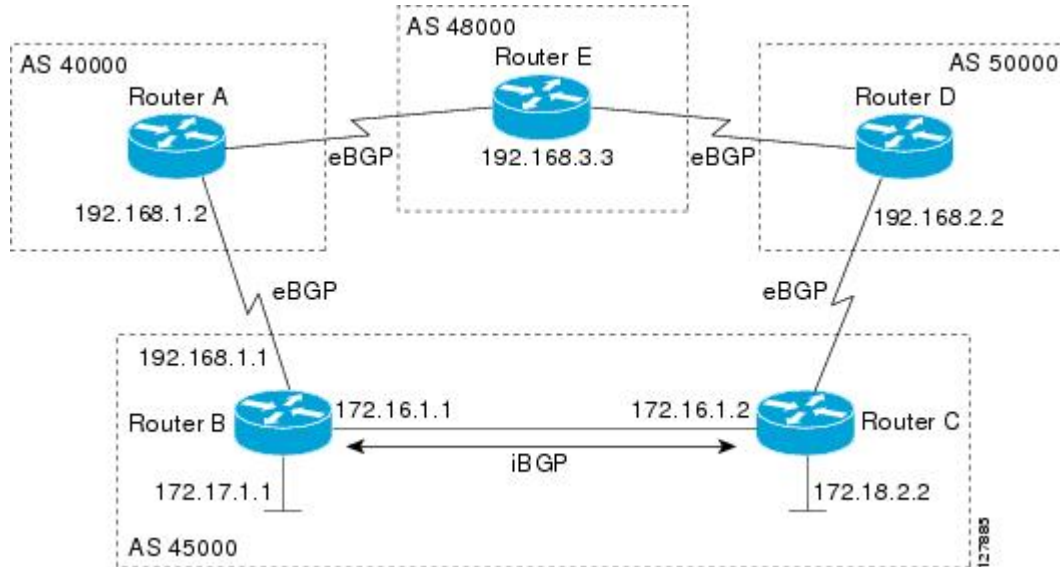
Influencing Outbound Path Selection Using the Local_Pref Attribute

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 23: Network Topology for Outbound Path Selection



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. **bgp default local-preference** *value*
6. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*][**route-map** *route-map-name*]
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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> Example: <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	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 5	bgp default local-preference <i>value</i> Example: <pre>Router(config-router)# bgp default local-preference 150</pre>	<p>Changes the default local preference value.</p> <ul style="list-style-type: none"> • 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 6	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	network <i>network-number</i> [mask <i>network-mask</i>][route-map <i>route-map-name</i>] Example: <pre>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • 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	Purpose
Step 8	neighbor {ip-address peer-group-name} activate Example: <pre>Router(config-router-af)# neighbor 192.168.1.2 activate</pre>	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: <pre>Router(config-router-af)# end</pre>	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: <pre>Router# show ip bgp 192.168.3.0 255.255.255.0</pre>	Displays the entries in the BGP routing table. <ul style="list-style-type: none"> 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Filtering Outbound BGP Route Prefixes

Perform this task to use BGP prefix-based outbound route filtering to influence outbound path selection.

Before You Begin

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. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **ebgp-multihop** [*hop-count*]
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
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

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] { deny <i>network/length</i> permit <i>network/length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: <pre>Router(config)# ip prefix-list FILTER seq 10 permit 192.168.1.0/24</pre>	Creates a prefix list for prefix-based outbound route filtering. <ul style="list-style-type: none"> • 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.
Step 4	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 100</pre>	Enters router configuration mode, and creates a BGP routing process.

	Command or Action	Purpose
Step 5	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.1.1.1 remote-as 200</pre>	<p>Establishes peering with the specified neighbor or peer group. BGP peering must be established before ORF capabilities can be exchanged.</p> <ul style="list-style-type: none"> The example establishes peering with the 10.1.1.1 neighbor.
Step 6	<p>neighbor <i>ip-address</i> ebgp-multihop [<i>hop-count</i>]</p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.1.1.1 ebgp-multihop</pre>	<p>Accepts or initiates BGP connections to external peers residing on networks that are not directly connected.</p>
Step 7	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. <p>Note Outbound route filtering is configured on a per-address family basis.</p>
Step 8	<p>neighbor <i>ip-address</i> capability orf prefix-list [send receive both]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.1.1 capability orf prefix-list both</pre>	<p>Enables the ORF capability on the local router, and enables ORF capability advertisement to the BGP peer specified with the <i>ip-address</i> argument.</p> <ul style="list-style-type: none"> 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.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} prefix-list <i>prefix-list-name</i> {in out}</p>	<p>Applies an inbound prefix-list filter to prevent distribution of BGP neighbor information.</p>

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in</pre>	<ul style="list-style-type: none"> 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.
Step 10	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode, and enters privileged EXEC mode.
Step 11	<p>clear ip bgp {ip-address *} in prefix-filter</p> <p>Example:</p> <pre>Router# clear ip bgp 10.1.1.1 in prefix-filter</pre>	<p>Clears BGP outbound route filters and initiates an inbound soft reset.</p> <ul style="list-style-type: none"> A single neighbor or all neighbors can be specified. <p>Note The inbound soft refresh must be initiated with the clear ip bgp command in order for this feature to function.</p>

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

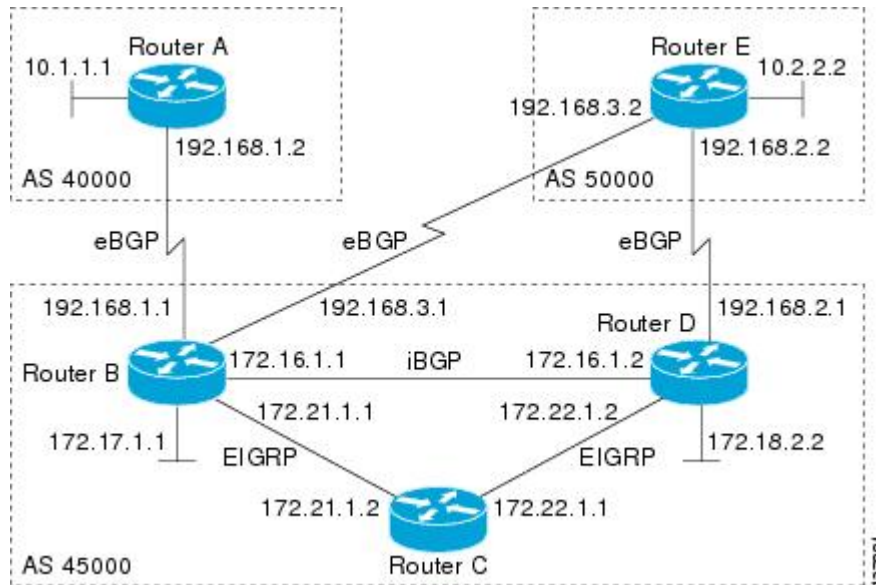
Perform this task to configure your network to access two ISPs where one ISP is the preferred route and the second ISP is a backup route. In the figure below Router B in autonomous system 45000 has BGP peers in two ISPs, autonomous system 40000 and autonomous system 50000. Using this task, Router B will 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.

**Note**

The weights assigned with the **set weight** route-map configuration command override the weights assigned using the **neighbor weight** command.

Figure 24: Multihoming with Two ISPs

**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*]
7. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
12. **end**
13. **clear ip bgp** {*** | *ip-address* | *peer-group-name*} [**soft** [**in** | **out**]]
14. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode, and creates a BGP routing process.</p>
Step 4	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	<p>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.</p>
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>network <i>network-number</i> [mask <i>network-mask</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} weight <i>number</i></p>	<p>Assigns a weight to a BGP peer connection.</p>

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 weight 150</pre>	<ul style="list-style-type: none"> In this example, the weight attribute for routes received from the BGP peer 192.168.1.2 is set to 150.
Step 8	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 9	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	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 10	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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. <p>The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</p>
Step 11	<p>neighbor <i>{ip-address peer-group-name}</i> weight <i>number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 weight 100</pre>	<p>Assigns a weight to a BGP peer connection.</p> <ul style="list-style-type: none"> In this example, the weight attribute for routes received from the BGP peer 192.168.3.2 is set to 100.
Step 12	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 13	<p>clear ip bgp {* <i>ip-address</i> <i>peer-group-name</i>} [soft [in out]]</p> <p>Example:</p> <pre>Router# clear ip bgp *</pre>	<p>(Optional) Clears BGP outbound route filters and initiates an outbound soft reset. A single neighbor or all neighbors can be specified.</p>
Step 14	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>]</p> <p>Example:</p> <pre>Router# show ip bgp</pre>	<p>Displays the entries in the BGP routing table.</p> <ul style="list-style-type: none"> 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Examples

The following example shows the BGP routing table at Router B with the weight attributes assigned to routes. The route through 192.168.1.2 (Router A in the figure above) has the highest weight attribute and will be the preferred route to network 10.3.0.0, wherein the network 10.3.0.0 is accessible through Router A and Router E. If this route (through Router B) fails for some reason, the route through 192.168.3.2 (Router E) will be used to reach network 10.3.0.0. This way, redundancy is provided for reaching Router B.

```
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

   Network        Next Hop           Metric    LocPrf  Weight    Path
*> 10.1.1.0/24    192.168.1.2         0          150     40000    i
*> 10.2.2.0/24    192.168.3.2         0          100     50000    i

*> 10.3.0.0/16    192.168.1.2         0          150     40000    i
*                 192.168.3.2         0          100     50000    i

*> 172.17.1.0/24  0.0.0.0             0          32768                   i
```

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. **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. Repeat Step 7 to apply another route map to the neighbor specified in Step 7.
9. **exit**
10. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
13. Repeat Step 10 to apply another route map to the neighbor specified in Step 10.
14. **exit**
15. **exit**
16. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
17. Repeat Step 14 to establish another static route.
18. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
19. **set metric** *value*
20. **exit**
21. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
22. **set metric** *value*
23. **exit**
24. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
25. **end**
26. **show ip route** [*ip-address*] [*mask*] [**longer-prefixes**]
27. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i> Example: <pre>Router(config-router)# neighbor 192.168.2.1 remote-as 45000</pre>	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. <ul style="list-style-type: none"> • In this example, the BGP peer at Router D is added to the BGP routing table.
Step 5	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: <pre>Router(config-router)# address-family ipv4 unicast</pre>	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] Example: <pre>Router(config-router-af)# network 10.2.2.0 mask 255.255.255.0</pre>	Specifies a network as local to this autonomous system and adds it to the BGP routing table. <ul style="list-style-type: none"> • 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	Purpose
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.1 route-map BLOCK in</pre> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.1 route-map SETMETRIC1 out</pre>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> In the first example, the route map named BLOCK is applied to inbound routes at Router E. In the second example, the route map named SETMETRIC1 is applied to outbound routes to Router D. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
Step 8	Repeat Step 7 to apply another route map to the neighbor specified in Step 7.	--
Step 9	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 10	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.1 remote-as 45000</pre>	<p>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.</p> <ul style="list-style-type: none"> In this example, the BGP peer at Router D is added to the BGP routing table.
Step 11	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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. <p>The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</p>
Step 12	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.1 route-map BLOCK in</pre>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> In the first example, the route map named BLOCK is applied to inbound routes at Router E. In the second example, the route map named SETMETRIC2 is applied to outbound routes to Router D.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.1 route-map SETMETRIC2 out</pre>	<p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
Step 13	Repeat Step 10 to apply another route map to the neighbor specified in Step 10.	--
Step 14	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 15	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 16	<p>ip route <i>prefix mask {ip-address interface-type interface-number [ip-address]} [distance] [name] [permanent track number] [tag tag]</i></p> <p>Example:</p> <pre>Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50</pre> <p>Example:</p> <pre>Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50</pre> <p>Example:</p> <p>and</p> <p>Example:</p> <pre>Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1 40</pre>	<p>Establishes a static route.</p> <ul style="list-style-type: none"> • 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. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
Step 17	Repeat Step 14 to establish another static route.	--
Step 18	<p>route-map <i>map-name [permit deny] [sequence-number]</i></p> <p>Example:</p> <pre>Router(config)# route-map SETMETRIC1 permit 10</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> • In this example, a route map named SETMETRIC1 is created.

	Command or Action	Purpose
Step 19	set metric <i>value</i> Example: Router(config-route-map)# set metric 100	Sets the MED metric value.
Step 20	exit Example: Router(config-route-map)# exit	Exits route map configuration mode and enters global configuration mode.
Step 21	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map SETMETRIC2 permit 10	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named SETMETRIC2 is created.
Step 22	set metric <i>value</i> Example: Router(config-route-map)# set metric 50	Sets the MED metric value.
Step 23	exit Example: Router(config-route-map)# exit	Exits route map configuration mode and enters global configuration mode.
Step 24	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map BLOCK deny 10	Configures a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, a route map named BLOCK is created to block all incoming routes from autonomous system 45000.
Step 25	end Example: Router(config-route-map)# end	Exits route map configuration mode and enters privileged EXEC mode.
Step 26	show ip route [<i>ip-address</i>] [<i>mask</i>] [<i>longer-prefixes</i>] Example: Router# show ip route	(Optional) Displays route information from the routing tables. <ul style="list-style-type: none"> Use this command at Router E in the figure above after Router B and Router D have received update information containing the MED metric from Router E.

	Command or Action	Purpose
		<ul style="list-style-type: none"> Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.
Step 27	<p>show ip bgp <i>[network]</i> <i>[network-mask]</i></p> <p>Example:</p> <pre>Router# show ip bgp 172.17.1.0 255.255.255.0</pre>	<p>(Optional) Displays the entries in the BGP routing table.</p> <ul style="list-style-type: none"> Use this command at Router E in the figure above after Router B and Router D have received update information containing the MED metric from Router E. Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.

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, Ethernet0/0
 C    192.168.2.0/24 is directly connected, Serial3/0
 C    192.168.3.0/24 is directly connected, Serial2/0
 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, * 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    0.0.0.0             0         32768 i
```

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
  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                 50        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 i

```

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
  Network          Next Hop             Metric LocPrf Weight Path
*> 10.2.2.0/24     192.168.2.2            100        0 50000 i
*> 172.16.1.0/24   0.0.0.0                 0         32768 i

```

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.



Note

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. **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*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
12. **exit**
13. **exit**
14. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
15. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
16. **match as-path** *path-list-number*
17. **end**
18. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 45000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	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 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>network <i>network-number</i> [mask <i>network-mask</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 7	<p>neighbor <i>{ip-address peer-group-name}</i> route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 route-map localonly out</pre>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> In this example, the route map named localonly is applied to outbound routes to Router A.
Step 8	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 9	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	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.

	Command or Action	Purpose
Step 10	<p>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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. <p>The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</p>
Step 11	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 route-map localonly out</pre>	<p>Applies a route map to incoming or outgoing routes.</p> <ul style="list-style-type: none"> In this example, the route map named localonly is applied to outbound routes to Router E.
Step 12	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
Step 13	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
Step 14	<p>ip as-path access-list <i>access-list-number</i> {deny permit} <i>as-regular-expression</i></p> <p>Example:</p> <pre>Router(config)# ip as-path access-list 10 permit ^\$</pre>	<p>Defines a BGP-related access list.</p> <ul style="list-style-type: none"> In this example, the access list number 10 is defined to permit only locally originated BGP routes.
Step 15	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map localonly permit 10</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, a route map named localonly is created.
Step 16	<p>match as-path <i>path-list-number</i></p>	<p>Matches a BGP autonomous system path access list.</p>

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-route-map)# match as-path 10</pre>	<ul style="list-style-type: none"> In this example, the BGP autonomous system path access list created in Step 12 is used for the match clause.
Step 17	<p>end</p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	Exits route map configuration mode and enters privileged EXEC mode.
Step 18	<p>show ip bgp [network] [network-mask]</p> <p>Example:</p> <pre>Router# show ip bgp</pre>	<p>Displays the entries in the BGP routing table.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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

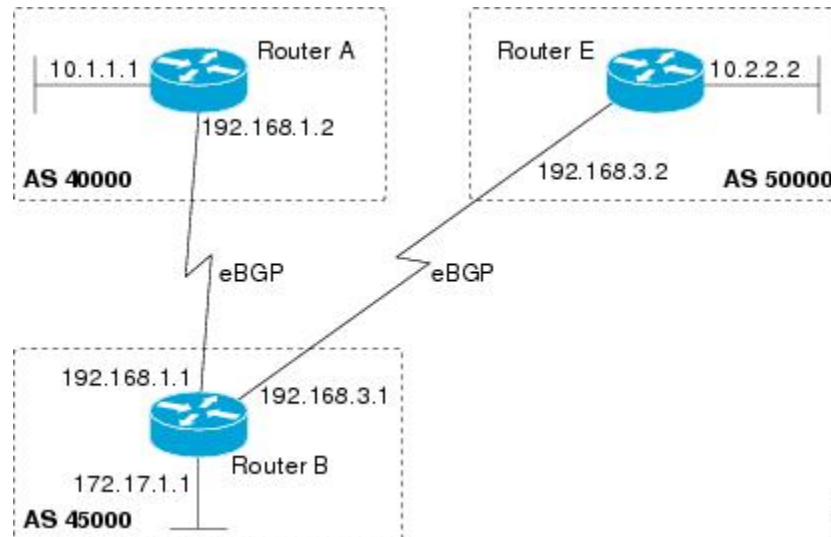
The tasks in this section help you configure BGP policies that filter the traffic in your BGP network. The following optional tasks demonstrate some of the various methods by which traffic can be filtered in your BGP network:

Filtering BGP Prefixes with Prefix Lists

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 25: BGP Topology for Configuring BGP Policies Tasks



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. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Repeat Step 5 for all BGP peers.
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **aggregate-address** *address mask* [**as-set**]
9. **neighbor** *ip-address* **prefix-list** *list-name* {**in** | **out**}
10. **exit**
11. **exit**
12. **ip prefix-list** *list-name* [**seq** *seq-number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] [**eq** *eq-value*]
13. **end**
14. **show ip prefix-list** [**detail** | **summary**] [*prefix-list-name* [**seq** *seq-number* | *network/length* [**longer** | **first-match**]]]
15. **clear ip prefix-list** {***** | *ip-address* | *peer-group-name*} **out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local router.</p>
Step 5	<p>Repeat Step 5 for all BGP peers.</p>	--
Step 6	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	<p>network <i>network-number</i> [mask <i>network-mask</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> • 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	Purpose
Step 8	aggregate-address <i>address mask</i> [as-set] Example: <pre>Router(config-router-af)# aggregate-address 172.0.0.0 255.0.0.0</pre>	Creates an aggregate entry in a BGP routing table. <ul style="list-style-type: none"> • 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. Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 9	neighbor <i>ip-address</i> prefix-list <i>list-name</i> { in out } Example: <pre>Router(config-router-af)# neighbor 192.168.1.2 prefix-list super172 out</pre>	Distributes BGP neighbor information as specified in a prefix list. <ul style="list-style-type: none"> • In this example, a prefix list called super172 is set for outgoing routes to Router A.
Step 10	exit Example: <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 11	exit Example: <pre>Router(config-router) exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 12	ip prefix-list <i>list-name</i> [seq <i>seq-number</i>] { deny <i>network/length</i> permit <i>network/length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] [eq <i>eq-value</i>] Example: <pre>Router(config)# ip prefix-list super172 permit 172.0.0.0/8</pre>	Defines a BGP-related prefix list and enters access list configuration mode. <ul style="list-style-type: none"> • 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.
Step 13	end Example: <pre>Router(config-access-list)# end</pre>	Exits access list configuration mode and enters privileged EXEC mode.
Step 14	show ip prefix-list [detail summary] [<i>prefix-list-name</i> [seq <i>seq-number</i> <i>network/length</i>] [longer first-match]]	Displays information about prefix lists. <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router# show ip prefix-list detail super172</pre>	
Step 15	<p>clear ip prefix-list <i>{* ip-address peer-group-name}</i> out</p> <p>Example:</p> <pre>Router# clear ip prefix-list super172 out</pre>	<p>Resets the hit count of the prefix list entries.</p> <ul style="list-style-type: none"> In this example, the hit count for the prefix list called super172 will be reset.

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 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 filter is enabled, traffic can be received from both Router A and Router C, but updates originating from autonomous system 50000 (Router C) are not forwarded by Router B to Router A. If any updates from Router C originated from another autonomous system, they would be forwarded because they would contain both autonomous system 50000 and another autonomous system number, and that would not match the AS-path access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip as-path access-list** *access-list-number* {deny | permit} *as-regular-expression*
4. Repeat Step 3 for all entries required in the AS-path access list.
5. **router bgp** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. Repeat Step 6 for all BGP peers.
8. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
9. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {in | out}
10. **end**
11. **show ip bgp regexp** *as-regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip as-path access-list <i>access-list-number</i> {deny permit} <i>as-regular-expression</i> Example: Device(config)# ip as-path access-list 100 deny ^50000\$ Example: Device(config)# ip as-path access-list 100 permit .*	Defines a BGP-related access list and enters access list configuration mode. • In the first example, access list number 100 is defined to deny any AS-path that starts and ends with 50000. • 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. Note Two examples are shown here because the task example requires both these statements to be configured.
Step 4	Repeat Step 3 for all entries required in the AS-path access list.	—

	Command or Action	Purpose
Step 5	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router.
Step 7	Repeat Step 6 for all BGP peers.	—
Step 8	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 unicast	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 9	neighbor { <i>ip-address</i> <i>peer-group-name</i> } filter-list <i>access-list-number</i> { in out } Example: Device(config-router-af)# neighbor 192.168.1.2 filter-list 100 out	Distributes BGP neighbor information as specified in a prefix list. <ul style="list-style-type: none"> • In this example, an access list number 100 is set for outgoing routes to Router A.
Step 10	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	show ip bgp regexp <i>as-regular-expression</i> Example: Device# show ip bgp regexp ^50000\$	Displays routes that match the regular expression. <ul style="list-style-type: none"> • To verify the regular expression, you can use this command. • In this example, all paths that match the expression “starts and ends with 50000” will be displayed.

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:

```
Device# 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, * 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             150 50000 i
```

Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers

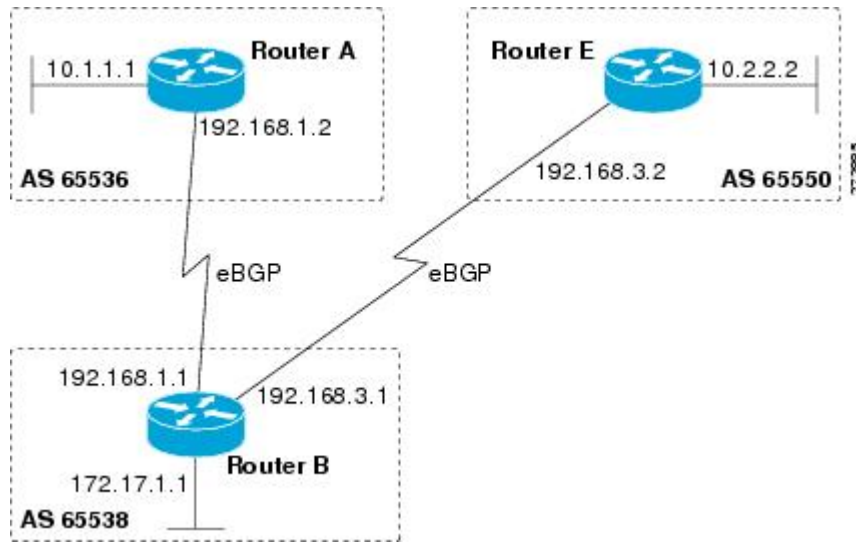
In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SX11, and later releases, BGP support for 4-octet (4-byte) autonomous system numbers was introduced. The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format, for example, Router B is in autonomous system number 65538 in the figure below. For more details about the introduction of 4-byte autonomous system numbers, see the “BGP Autonomous System Number Formats” section.

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 Releases 12.0(22)S, 12.2(15)T, 12.2(18)S, 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 26: 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. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. Repeat Step 4 for all BGP peers.
6. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**}
9. **exit**
10. **exit**
11. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
12. Repeat Step 11 for all entries required in the AS-path access list.
13. **end**
14. **show ip bgp regexp** *as-regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 65538</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: <pre>Router(config-router-af)# neighbor 192.168.1.2 remote-as 65536</pre>	Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router. <ul style="list-style-type: none"> • In this example, the IP address for the neighbor at Router A is added.
Step 5	Repeat Step 4 for all BGP peers.	--
Step 6	address-family ipv4 [unicast multicast] vrf <i>vrf-name</i> Example: <pre>Router(config-router)# address-family ipv4 unicast</pre>	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	network <i>network-number</i> [mask <i>network-mask</i>] Example: <pre>Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. <ul style="list-style-type: none"> • For exterior protocols the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates. <p>Note Only partial syntax is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

	Command or Action	Purpose
Step 8	<p>neighbor <i>{ip-address peer-group-name}</i> filter-list <i>access-list-number</i>{in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 filter-list 99 out</pre>	<p>Distributes BGP neighbor information as specified in a prefix list.</p> <ul style="list-style-type: none"> In this example, an access list number 99 is set for outgoing routes to Router A.
Step 9	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and returns to router configuration mode.</p>
Step 10	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and returns to global configuration mode.</p>
Step 11	<p>ip as-path access-list <i>access-list-number</i> {deny permit} <i>as-regular-expression</i></p> <p>Example:</p> <pre>Router(config)# ip as-path access-list 99 deny ^65550\$</pre> <p>Example:</p> <pre>and</pre> <p>Example:</p> <pre>Router(config)# ip as-path access-list 99 permit .*</pre>	<p>Defines a BGP-related access list and enters access list configuration mode.</p> <ul style="list-style-type: none"> In the first example, access list number 99 is defined to deny any AS-path that starts and ends with 65550. 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. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
Step 12	<p>Repeat Step 11 for all entries required in the AS-path access list.</p>	--
Step 13	<p>end</p> <p>Example:</p> <pre>Router(config-access-list)# end</pre>	<p>Exits access list configuration mode and returns to privileged EXEC mode.</p>
Step 14	<p>show ip bgp regexp <i>as-regular-expression</i></p> <p>Example:</p> <pre>Router# show ip bgp regexp ^65550\$</pre>	<p>Displays routes that match the regular expression.</p> <ul style="list-style-type: none"> To verify the regular expression, you can use this command. In this example, all paths that match the expression "starts and ends with 65550" will be displayed.

Command or Action	Purpose
-------------------	---------

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:

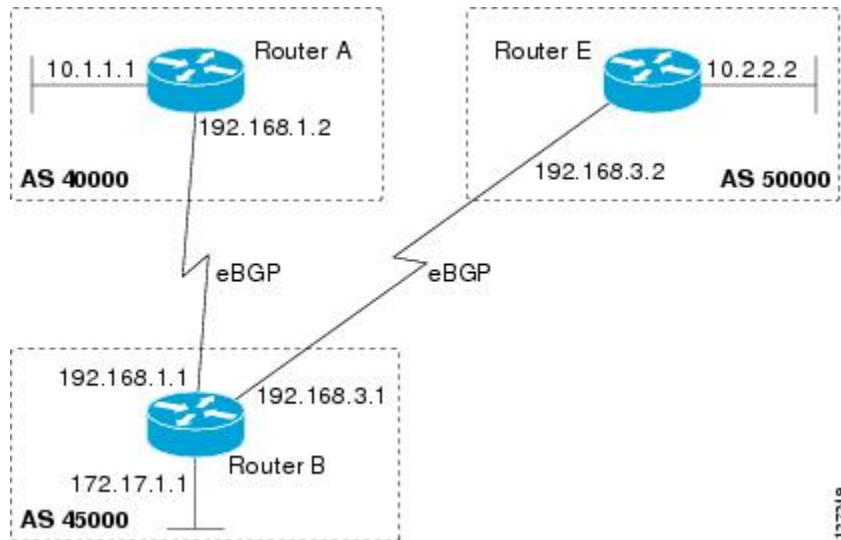
```
RouterB# show ip bgp regexp ^65550$
BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network             Next Hop         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

Filtering Traffic Using Community Lists

Perform this task to filter traffic by creating a BGP community list, referencing the community list within a route map, and then applying the route map to a neighbor.

In this task, Router B in the figure below is configured with route maps and a community list to control incoming routes.

Figure 27: Topology for Which a Community List Is Configured



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. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
11. **set weight** *weight*
12. **exit**
13. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
14. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
15. **set community** *community-number*
16. **exit**
17. **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*}
18. Repeat Step 17 to create all the required community lists.
19. **exit**
20. **show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 4	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>route-map-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.3.2 route-map 2000 in</pre>	<p>Applies a route map to inbound or outbound routes.</p> <ul style="list-style-type: none"> • In this example, the route map called 2000 is applied to inbound routes from the BGP peer at 192.168.3.2.
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 9	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p>	<p>Creates a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> • In this example, the route map called 2000 is defined.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config)# route-map 2000 permit 10</pre>	
Step 10	<p>match community {<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i> [exact]}</p> <p>Example:</p> <pre>Device(config-route-map)# match community 1</pre>	<p>Matches on the communities in a BGP community list.</p> <ul style="list-style-type: none"> In this example, the route's community attribute is matched to communities in community list 1.
Step 11	<p>set weight <i>weight</i></p> <p>Example:</p> <pre>Device(config-route-map)# set weight 30</pre>	<p>Sets the weight of BGP routes that match the community list.</p> <ul style="list-style-type: none"> In this example, any route that matches community list 1 will have its weight set to 30.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 13	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map 3000 permit 10</pre>	<p>Creates a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, the route map called 3000 is defined.
Step 14	<p>match community {<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i> [exact]}</p> <p>Example:</p> <pre>Device(config-route-map)# match community 2</pre>	<p>Matches on the communities in a BGP community list.</p> <ul style="list-style-type: none"> In this example, the route's COMMUNITIES attribute is matched to communities in community list 2.
Step 15	<p>set community <i>community-number</i></p> <p>Example:</p> <pre>Device(config-route-map)# set community 99</pre>	<p>Sets the BGP communities attribute.</p> <ul style="list-style-type: none"> In this example, any route that matches community list 2 will have the COMMUNITIES attribute set to 99.
Step 16	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 17	<p>ip community-list {<i>standard-list-number</i> standard <i>list-name</i> {deny permit} [<i>community-number</i>]</p>	<p>Creates a community list for BGP and controls access to it.</p>

	Command or Action	Purpose
	<p>[AA:NN] [internet] [local-AS] [no-advertise] [no-export]} {expanded-list-number expanded list-name {deny permit} regular-expression}</p> <p>Example:</p> <pre>Device(config)# ip community-list 1 permit 100</pre> <p>Example:</p> <pre>Device(config)# ip community-list 2 permit internet</pre>	<ul style="list-style-type: none"> In the first example, community list 1 permits routes with a COMMUNITIES attribute of 100. Router E routes all have a COMMUNITIES attribute of 100, so their weight will be set to 30. In the second example, community list 2 effectively permits all routes by specifying the internet community. 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. <p>Note Two examples are shown here because the task example requires both of these statements to be configured.</p>
Step 18	Repeat Step 17 to create all the required community lists.	—
Step 19	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	Exits global configuration mode and enters privileged EXEC mode.
Step 20	<p>show ip community-list [standard-list-number expanded-list-number community-list-name] [exact-match]</p> <p>Example:</p> <pre>Device# show ip community-list 1</pre>	Displays configured BGP community list entries.

Examples

The following sample output verifies that community list 1 has been created and it permits routes that have a community attribute of 100:

```
Device# show ip community-list 1
Community standard list 1
  permit 100
```

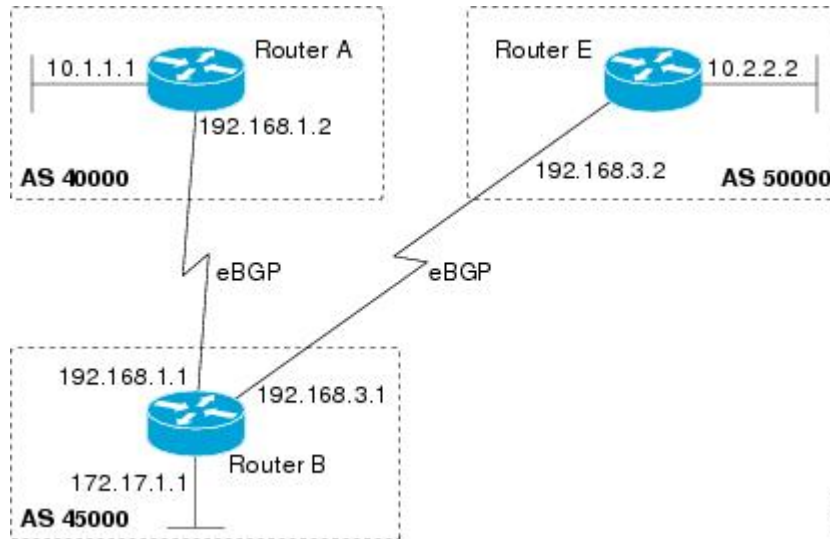
The following sample output verifies that community list 2 has been created and it effectively permits all routes by specifying the **internet** community:

```
Device# show ip community-list 2
Community standard list 2
  permit internet
```

Filtering Traffic Using Extended Community Lists

Perform this task to filter traffic by creating an extended BGP community list to control outbound routes.

Figure 28: Topology for Which a Community List Is Configured



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.



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, 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. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. Repeat the prior step for all of the required BGP peers.
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **network** *network-number* [**mask** *network-mask*]
13. **end**
14. **show ip extcommunity-list** [*list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip extcommunity-list { <i>expanded-list-number</i> expanded <i>list-name</i> <i>standard-list-number</i> standard <i>list-name</i> }	Enters IP extended community-list configuration mode to create or configure an extended community list. • In this example, the expanded community list DENY50000 is created.
Step 4	[<i>sequence-number</i>] { deny [<i>regular-expression</i>] exit permit [<i>regular-expression</i>]}	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.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-extcomm-list)# 10 deny _50000_</pre> <p>Example:</p> <pre>Device(config-extcomm-list)# 20 deny ^50000 .*</pre>	<ul style="list-style-type: none"> 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. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 5	Repeat Step 4 for all the required permit or deny entries in the extended community list.	—
Step 6	<p>resequence [<i>starting-sequence</i>] [<i>sequence-increment</i>]</p> <p>Example:</p> <pre>Device(config-extcomm-list)# resequence 50 100</pre>	<p>Resequences expanded community list entries.</p> <ul style="list-style-type: none"> 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-extcomm-list)# exit</pre>	Exits expanded community-list configuration mode and enters global configuration mode.
Step 8	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.
Step 10	Repeat the prior step for all of the required BGP peers.	—
Step 11	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]	Specifies the IPv4 address family and enters address family configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<ul style="list-style-type: none"> 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 in the address-family ipv4 command. The multicast keyword specifies IPv4 multicast address prefixes. <p>Note The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</p>
Step 12	<p>network <i>network-number</i> [mask <i>network-mask</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> For exterior protocols, the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
Step 14	<p>show ip extcommunity-list [<i>list-name</i>]</p> <p>Example:</p> <pre>Device# show ip extcommunity-list DENY50000</pre>	<p>Displays configured BGP expanded community list entries.</p>

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:

```
Device# show ip extcommunity-list DENY50000

Expanded extended community-list DENY50000
 50 deny _50000_
150 deny ^50000 .*
```

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.

Before You Begin

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 IPv6 match clauses in policy lists.
 - Policy lists are not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(22)S and 12.2(15)T. Reloading a router that is running an older version of Cisco IOS software may cause some routing policy configurations to be lost.
 - 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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip policy-list <i>policy-list-name</i> { permit deny } Example: Router(config)# ip policy-list POLICY-LIST-NAME-1 permit	Enters policy list configuration mode and creates a BGP policy list that will permit routes that are allowed by the match clauses that follow.
Step 4	match as-path <i>as-number</i> Example: Router(config-policy-list)# match as-path 500	Creates a match clause to permit routes from the specified autonomous system path.

	Command or Action	Purpose
Step 5	match metric <i>metric</i> Example: Router(config-policy-list)# match metric 10	Creates a match clause to permit routes with the specified metric.
Step 6	exit Example: Router(config-policy-list)# exit	Exits policy list configuration mode and enters global configuration mode.
Step 7	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map MAP-NAME-1 permit 10	Creates a route map and enters route map configuration mode.
Step 8	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [... <i>access-list-number</i> ... <i>access-list-name</i>] Example: Router(config-route-map)# match ip address 1	Creates a match clause to permit routes that match the specified <i>access-list-number</i> or <i>access-list-name</i> argument.
Step 9	match policy-list <i>policy-list-name</i> Example: Router(config-route-map)# match policy-list POLICY-LIST-NAME-1	Creates a clause that will match the specified policy list. <ul style="list-style-type: none"> • All match clauses within the policy list will be evaluated and processed. Multiple policy lists can be referenced with this command. • This command also supports AND or OR semantics like a standard match clause.
Step 10	set community <i>community-number</i> [additive] [<i>well-known-community</i>] none } Example: Router(config-route-map)# set community 10:1	Creates a clause to set or remove the specified community.
Step 11	set local-preference <i>preference-value</i> Example: Router(config-route-map)# set local-preference 140	Creates a clause to set the specified local preference value.

	Command or Action	Purpose
Step 12	end Example: Router(config-route-map) # end	Exits route map configuration mode and enters privileged EXEC mode.
Step 13	show ip policy-list [<i>policy-list-name</i>] Example: Router# show ip policy-list POLICY-LIST-NAME-1	Display information about configured policy lists and policy list entries.
Step 14	show route-map [<i>route-map-name</i>] Example: Router# show route-map	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.

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
  Match clauses:
    IP Policy lists:
      POLICY-LIST-NAME-1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

Filtering Traffic Using Continue Clauses in a BGP Route Map

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. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
10. **match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*]
11. **set community** { { [*community-number*] [*well-known-community*] [**additive**] } | **none** }
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode, and creates a BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 10.0.0.1 remote-as 50000	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 device.

	Command or Action	Purpose
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the device is placed in address family configuration mode for the IPv4 unicast address family if the unicast keyword is not specified. • The multicast keyword specifies IPv4 multicast address prefixes. • The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in</pre>	<p>Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
Step 9	<p>route-map <i>map-name</i> {permit deny}</p> <p>[<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map ROUTE-MAP-NAME permit 10</pre>	<p>Enters route-map configuration mode to create or configure a route map.</p>
Step 10	<p>match ip address {<i>access-list-number</i> <i>access-list-name</i>} [... <i>access-list-number</i> ... <i>access-list-name</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 1</pre>	<p>Configures a match command that specifies the conditions under which policy routing and route filtering occur.</p> <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
		Note The match and set commands used in this task are examples that are used to help describe the operation of the continue command. For a list of specific match and set commands, see the continue command in the <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 11	set community { { [community-number] [well-known-community] [additive]} none} Example: <pre>Device(config-route-map)# set community 10:1</pre>	Configures a set command that specifies the routing action to perform if the criteria enforced by the match commands are met. <ul style="list-style-type: none"> • Multiple set commands can be configured. • In this example, a clause is created to set the specified community number in aa:nn format.
Step 12	continue [sequence-number] Example: <pre>Device(config-route-map)# continue</pre>	Configures a route map to continue to evaluate and execute match statements after a successful match occurs. <ul style="list-style-type: none"> • If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. • 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 “implied continue.”
Step 13	end Example: <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	show route-map [map-name] Example: <pre>Device# show route-map</pre>	(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.

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.

```
Device# 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:
  Continue: 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

Example: Influencing Inbound Path Selection

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 route map named SET-LOCAL-PREF 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
```

Example: Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers

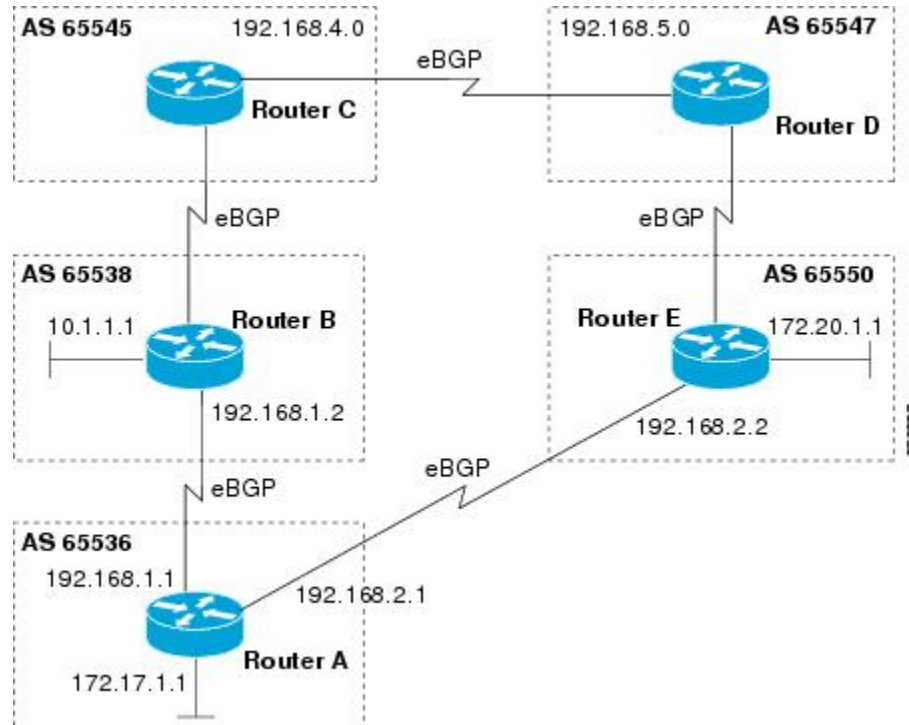
This example shows how to configure BGP to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXII, and later releases, BGP support for 4-octet (4-byte) autonomous system numbers 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 the “BGP Autonomous System Number Formats” section.

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 29: 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
```

Example: Filtering BGP Prefixes with Prefix Lists

This section contains the following examples:

Example: 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
!
route-map default-condition permit 10
 match ip address prefix-list cond
!
router 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:

```
router 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 Gigabit Ethernet interface 0/0/0:

```
router bgp 103
 distribute-list prefix name1 gateway name2 in gigabitethernet 0/0/0
```

Example: 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
```

Example: 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
```

Example: Filtering Traffic Using COMMUNITIES Attributes

This section contains two examples of the use of BGP COMMUNITIES attributes with route maps.

The first example configures a route map named *set-community*, which is applied to the outbound updates to the neighbor 172.16.232.50. The routes that pass access list 1 are given the well-known COMMUNITIES attribute value **no-export**. The remaining routes are advertised normally. The **no-export** community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```
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 configures a route map named *set-community*, which is applied to the outbound updates to neighbor 172.16.232.90. All the routes that originate from autonomous system 70 have the COMMUNITIES attribute values 200 200 added to their already existing communities. All other routes are advertised as normal.

```
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 70$
ip as-path access-list 2 permit .*

```

Example: Filtering Traffic Using AS-Path Filters

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.

```

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 3 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*

```

Example: Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SX11, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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.

```

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 Release 12.0(32)S12, and 12.4(24)T

The following example available in Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases 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.

**Note**

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

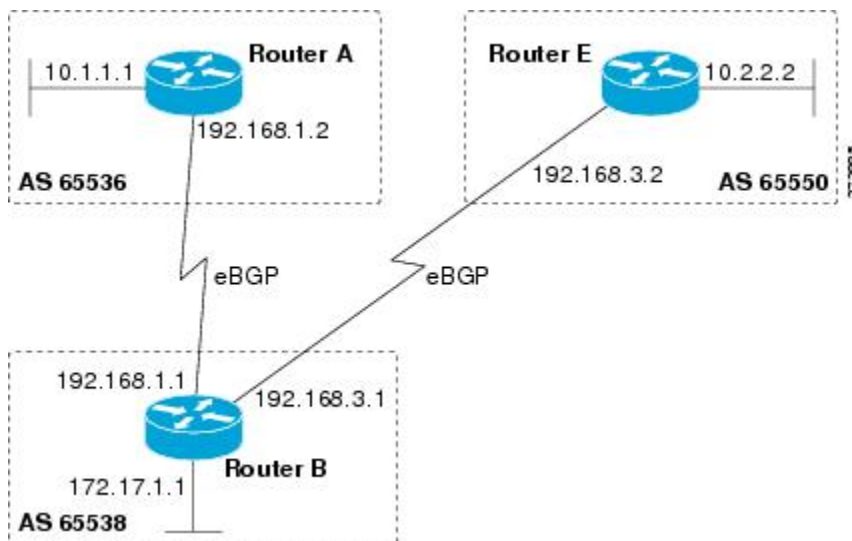
```
ip as-path access-list 2 permit ^1\.0$
router bgp 1.2
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.3.2 filter-list 2 in
end
```

Example: Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SX11, 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 Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SX11, 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 30: BGP Topology for Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers in Asplain Format



**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.

In this exam the figure above is configured with an extended named community list to specify that the BGP peer at 192.1681.2 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.

```
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
```

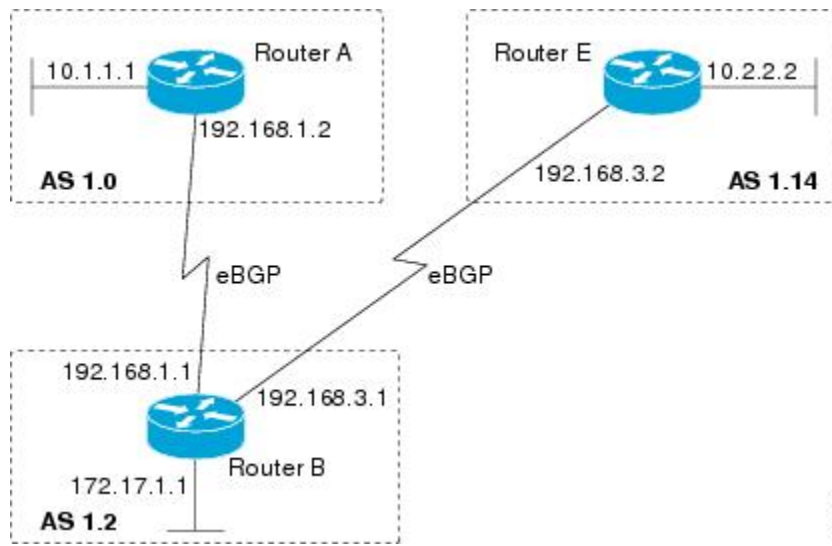
Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, 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 Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SX11, and later releases, this example works if you have configured `asdot` as the default display format using the `bgp asnotation dot` command.

Figure 31: BGP Topology for Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers in Asdot Format

**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.

In this exam 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.

```
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
```

Example: Filtering Traffic Using a BGP Route Map

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:

```
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
```

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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module
Configuring basic BGP tasks	“Configuring a Basic BGP Network” module
BGP fundamentals and description	<i>Large-Scale IP Network Solutions</i> , Khalid Raza and Mark Turner, Cisco Press, 2000

Related Topic	Document Title
Implementing and controlling BGP in scalable networks	<i>Building Scalable Cisco Networks</i> , Catherine Paquet and Diane Teare, Cisco Press, 2001
Interdomain routing basics	<i>Internet Routing Architectures</i> , Bassam Halabi, Cisco Press, 1997

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	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

RFCs

RFC	Title
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>
RFC 1774	<i>BGP-4 Protocol Analysis</i>
RFC 1930	<i>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>

RFC	Title
RFC 4684	<i>Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)</i>
RFC 4893	<i>BGP Support for Four-Octet AS Number Space</i>
RFC 5291	<i>Outbound Route Filtering Capability for BGP-4</i>
RFC 5396	<i>Textual Representation of Autonomous system (AS) Numbers</i>
RFC 5398	<i>Autonomous System (AS) Number Reservation for Documentation Use</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 25: Feature Information for Connecting to a Service Provider Using External BGP

Feature Name	Releases	Feature Configuration Information
BGP Increased Support of Numbered AS-Path Access Lists to 500	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	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.
BGP Named Community Lists	12.2(8)T 12.2(14)S 15.0(1)S	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.

Feature Name	Releases	Feature Configuration Information
BGP Route-Map Policy List Support	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	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.
BGP Support for Named Extended Community Lists	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	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.
BGP Support for Sequenced Entries in Extended Community Lists	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	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.
BGP 4 Prefix Filter and Inbound Route Maps	Cisco IOS XE 3.1.0SG	



BGP Named Community Lists

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.

- [Finding Feature Information, page 307](#)
- [Information About BGP Named Community Lists, page 307](#)
- [How to Use BGP Named Community Lists, page 308](#)
- [Configuration Examples for BGP Named Community Lists, page 318](#)
- [Additional References for BGP Named Community Lists, page 318](#)
- [Feature Information for BGP Named Community Lists, page 319](#)

Finding Feature Information

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

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

Information About BGP Named Community Lists

BGP COMMUNITIES Attribute

A BGP community is a group of routes that share a common property, regardless of their network, autonomous system, or any physical boundaries. In large networks, applying a common routing policy by using prefix lists or access lists requires individual peer statements on each networking device. Using the BGP COMMUNITIES attribute, BGP speakers with common routing policies can implement inbound or outbound route filters based on the community tag, rather than consult long lists of individual permit or deny statements. A COMMUNITIES attribute can contain multiple communities.

A route can belong to multiple communities. The network administrator defines the communities to which a route belongs. By default, all routes belong to the general Internet community.

In addition to numbered communities, there are several predefined (well-known) communities:

- no-export—Do not advertise this route to external BGP peers.
- no-advertise—Do not advertise this route to any peer.
- internet—Advertise this route to the Internet community. All BGP-speaking networking devices belong to this community.
- local-as—Do not send this route outside the local autonomous system.
- gshut—Community of routes gracefully shut down.

The COMMUNITIES 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 else the COMMUNITIES attribute will be discarded. By default, no COMMUNITIES attribute is sent to a neighbor. In order for a COMMUNITIES attribute to be sent to a neighbor, use the **neighbor send-community** command.

BGP Community Lists

A BGP community list is used to create groups of communities to use in a match clause of a route map. A community list can be used to control which routes are accepted, preferred, distributed, or advertised, for example. You can also use a community list to set, append, or modify the communities of a route.

- Standard community lists are used to specify well-known communities and community numbers.
- Expanded community lists are used to filter communities using a regular expression. Regular expressions are used to specify patterns to match COMMUNITIES attributes.

A BGP named community list allows you to assign a meaningful name to a community list. A named community list can be configured using community numbers, well-known communities, or regular expressions. All the rules of numbered community lists apply to named community lists, except that there is no limit on the number of named community lists that can be configured.

**Note**

A maximum of 100 standard community lists and 100 expanded community lists can be configured. A named community list does not have this limitation.

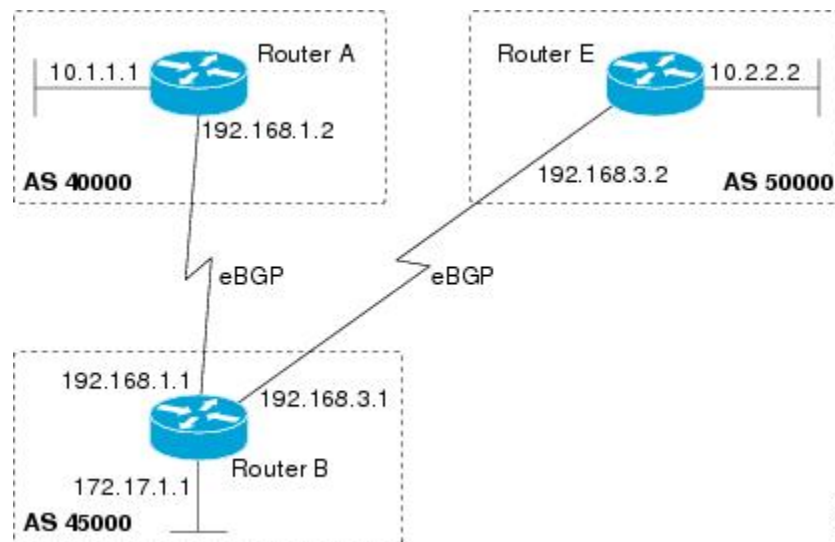
How to Use BGP Named Community Lists

Filtering Traffic Using Community Lists

Perform this task to filter traffic by creating a BGP community list, referencing the community list within a route map, and then applying the route map to a neighbor.

In this task, Router B in the figure below is configured with route maps and a community list to control incoming routes.

Figure 32: Topology for Which a Community List Is Configured



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. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
11. **set weight** *weight*
12. **exit**
13. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
14. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
15. **set community** *community-number*
16. **exit**
17. **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*}
18. Repeat Step 17 to create all the required community lists.
19. **exit**
20. **show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.3.2 remote-as 50000	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.
Step 5	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 unicast	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-map <i>route-map-name</i> { in out } Example: Device(config-router-af)# neighbor 192.168.3.2 route-map 2000 in	Applies a route map to inbound or outbound routes. <ul style="list-style-type: none"> • In this example, the route map called 2000 is applied to inbound routes from the BGP peer at 192.168.3.2.
Step 7	exit Example: Device(config-router-af)# exit	Exits address family configuration mode and enters router configuration mode.
Step 8	exit Example: Device(config-router)# exit	Exits router configuration mode and enters global configuration mode.
Step 9	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]	Creates a route map and enters route map configuration mode. <ul style="list-style-type: none"> • In this example, the route map called 2000 is defined.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config)# route-map 2000 permit 10</pre>	
Step 10	<p>match community {<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i> [exact]}</p> <p>Example:</p> <pre>Device(config-route-map)# match community 1</pre>	<p>Matches on the communities in a BGP community list.</p> <ul style="list-style-type: none"> In this example, the route's community attribute is matched to communities in community list 1.
Step 11	<p>set weight <i>weight</i></p> <p>Example:</p> <pre>Device(config-route-map)# set weight 30</pre>	<p>Sets the weight of BGP routes that match the community list.</p> <ul style="list-style-type: none"> In this example, any route that matches community list 1 will have its weight set to 30.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 13	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map 3000 permit 10</pre>	<p>Creates a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, the route map called 3000 is defined.
Step 14	<p>match community {<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i> [exact]}</p> <p>Example:</p> <pre>Device(config-route-map)# match community 2</pre>	<p>Matches on the communities in a BGP community list.</p> <ul style="list-style-type: none"> In this example, the route's COMMUNITIES attribute is matched to communities in community list 2.
Step 15	<p>set community <i>community-number</i></p> <p>Example:</p> <pre>Device(config-route-map)# set community 99</pre>	<p>Sets the BGP communities attribute.</p> <ul style="list-style-type: none"> In this example, any route that matches community list 2 will have the COMMUNITIES attribute set to 99.
Step 16	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
Step 17	<p>ip community-list {<i>standard-list-number</i> standard <i>list-name</i> {deny permit} [<i>community-number</i>]</p>	<p>Creates a community list for BGP and controls access to it.</p>

	Command or Action	Purpose
	<p>[AA:NN] [internet] [local-AS] [no-advertise] [no-export]} {expanded-list-number expanded list-name {deny permit} regular-expression}</p> <p>Example:</p> <pre>Device(config)# ip community-list 1 permit 100</pre> <p>Example:</p> <pre>Device(config)# ip community-list 2 permit internet</pre>	<ul style="list-style-type: none"> In the first example, community list 1 permits routes with a COMMUNITIES attribute of 100. Router E routes all have a COMMUNITIES attribute of 100, so their weight will be set to 30. In the second example, community list 2 effectively permits all routes by specifying the internet community. 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. <p>Note Two examples are shown here because the task example requires both of these statements to be configured.</p>
Step 18	Repeat Step 17 to create all the required community lists.	—
Step 19	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	Exits global configuration mode and enters privileged EXEC mode.
Step 20	<p>show ip community-list [standard-list-number expanded-list-number community-list-name] [exact-match]</p> <p>Example:</p> <pre>Device# show ip community-list 1</pre>	Displays configured BGP community list entries.

Examples

The following sample output verifies that community list 1 has been created and it permits routes that have a community attribute of 100:

```
Device# show ip community-list 1
Community standard list 1
  permit 100
```

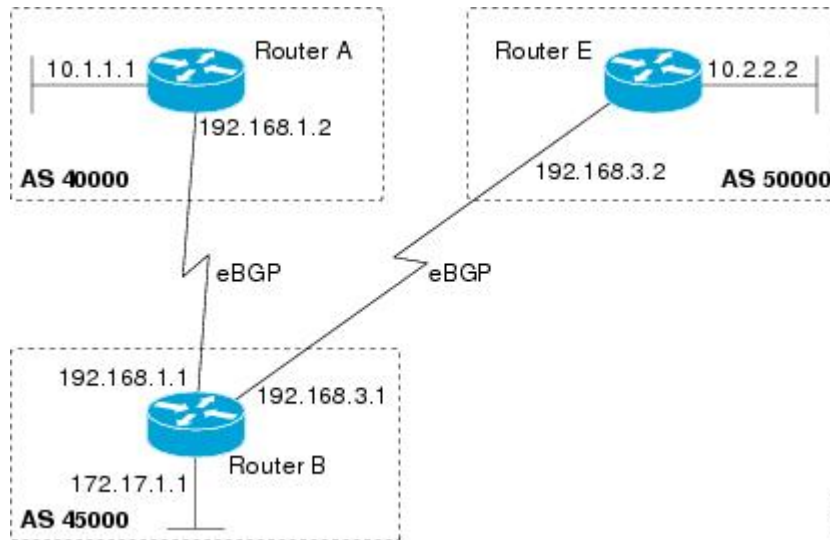
The following sample output verifies that community list 2 has been created and it effectively permits all routes by specifying the **internet** community:

```
Device# show ip community-list 2
Community standard list 2
  permit internet
```

Filtering Traffic Using Extended Community Lists

Perform this task to filter traffic by creating an extended BGP community list to control outbound routes.

Figure 33: Topology for Which a Community List Is Configured



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.



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, 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. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. Repeat the prior step for all of the required BGP peers.
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **network** *network-number* [**mask** *network-mask*]
13. **end**
14. **show ip extcommunity-list** [*list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip extcommunity-list { <i>expanded-list-number</i> expanded <i>list-name</i> <i>standard-list-number</i> standard <i>list-name</i> }	Enters IP extended community-list configuration mode to create or configure an extended community list. • In this example, the expanded community list DENY50000 is created.
Step 4	[<i>sequence-number</i>] { deny [<i>regular-expression</i>] exit permit [<i>regular-expression</i>]}	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.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-extcomm-list)# 10 deny _50000_</pre> <p>Example:</p> <pre>Device(config-extcomm-list)# 20 deny ^50000 .*</pre>	<ul style="list-style-type: none"> 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. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 5	Repeat Step 4 for all the required permit or deny entries in the extended community list.	—
Step 6	<p>resequence <i>[starting-sequence]</i> <i>[sequence-increment]</i></p> <p>Example:</p> <pre>Device(config-extcomm-list)# resequence 50 100</pre>	<p>Resequences expanded community list entries.</p> <ul style="list-style-type: none"> 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-extcomm-list)# exit</pre>	Exits expanded community-list configuration mode and enters global configuration mode.
Step 8	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 9	<p>neighbor <i>{ip-address peer-group-name}</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.
Step 10	Repeat the prior step for all of the required BGP peers.	—
Step 11	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]	Specifies the IPv4 address family and enters address family configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<ul style="list-style-type: none"> 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 in the address-family ipv4 command. The multicast keyword specifies IPv4 multicast address prefixes. <p>Note The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.</p>
Step 12	<p>network <i>network-number</i> [mask <i>network-mask</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0</pre>	<p>(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table.</p> <ul style="list-style-type: none"> For exterior protocols, the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
Step 14	<p>show ip extcommunity-list [<i>list-name</i>]</p> <p>Example:</p> <pre>Device# show ip extcommunity-list DENY50000</pre>	<p>Displays configured BGP expanded community list entries.</p>

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:

```
Device# show ip extcommunity-list DENY50000

Expanded extended community-list DENY50000
 50 deny _50000_
150 deny ^50000 .*
```

Configuration Examples for BGP Named Community Lists

Example: Filtering Traffic Using COMMUNITIES Attributes

This section contains two examples of the use of BGP COMMUNITIES attributes with route maps.

The first example configures a route map named *set-community*, which is applied to the outbound updates to the neighbor 172.16.232.50. The routes that pass access list 1 are given the well-known COMMUNITIES attribute value **no-export**. The remaining routes are advertised normally. The **no-export** community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```
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 configures a route map named *set-community*, which is applied to the outbound updates to neighbor 172.16.232.90. All the routes that originate from autonomous system 70 have the COMMUNITIES attribute values 200 200 added to their already existing communities. All other routes are advertised as normal.

```
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 70$
ip as-path access-list 2 permit .*
```

Additional References for BGP Named Community Lists

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1997	<i>BGP Communities Attribute</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Named Community Lists

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 26: Feature Information for BGP Named Community Lists

Feature Name	Releases	Feature Information
BGP Named Community Lists		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.



CHAPTER 10

BGP Prefix-Based Outbound Route Filtering

The BGP Prefix-Based Outbound Route Filtering (ORF) 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.

- [Finding Feature Information, page 321](#)
- [Information About BGP Prefix-Based Outbound Route Filtering, page 321](#)
- [How to Configure BGP Prefix-Based Outbound Route Filtering, page 322](#)
- [Configuration Examples for BGP Prefix-Based Outbound Route Filtering, page 325](#)
- [Additional References, page 326](#)
- [Feature Information for BGP Prefix-Based Outbound Route Filtering, page 327](#)

Finding Feature Information

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

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

Information About BGP Prefix-Based Outbound Route Filtering

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.

How to Configure BGP Prefix-Based Outbound Route Filtering

Filtering Outbound Routes Based on BGP Prefix

Before You Begin

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] { deny <i>network / length</i> permit <i>network / length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: Device(config)# ip prefix-list FILTER seq 10 permit 192.168.1.0/24	Creates a prefix list for prefix-based outbound route filtering. <ul style="list-style-type: none"> • 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.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Enters router configuration mode, and creates a BGP routing process.

	Command or Action	Purpose
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. <p>Note Outbound route filtering is configured on a per-address family basis.</p>
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.1.1.1 remote-as 200</pre>	<p>Establishes peering with the specified neighbor or peer group. BGP peering must be established before ORF capabilities can be exchanged.</p> <ul style="list-style-type: none"> The example establishes peering with the 10.1.1.1 neighbor.
Step 7	<p>neighbor <i>ip-address</i> ebgp-multihop [<i>hop-count</i>]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.1.1.1 ebgp-multihop</pre>	<p>Accepts or initiates BGP connections to external peers residing on networks that are not directly connected.</p>
Step 8	<p>neighbor <i>ip-address</i> capability orf prefix-list [send receive both]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.1.1.1 capability orf prefix-list both</pre>	<p>Enables the ORF capability on the local router, and enables ORF capability advertisement to the BGP peer specified with the <i>ip-address</i> argument.</p> <ul style="list-style-type: none"> 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.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>prefix-list <i>prefix-list-name</i> {in out}</p>	<p>Applies an inbound prefix-list filter to prevent distribution of BGP neighbor information.</p>

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in</pre>	<ul style="list-style-type: none"> 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.
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode, and enters privileged EXEC mode.
Step 11	<p>clear ip bgp {ip-address *} in prefix-filter</p> <p>Example:</p> <pre>Device# clear ip bgp 10.1.1.1 in prefix-filter</pre>	<p>Clears BGP outbound route filters and initiates an inbound soft reset.</p> <ul style="list-style-type: none"> A single neighbor or all neighbors can be specified. <p>Note The inbound soft refresh must be initiated with the clear ip bgp command in order for this feature to function.</p>

Configuration Examples for BGP Prefix-Based Outbound Route Filtering

Example: Influencing Outbound Path Selection

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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Prefix-Based Outbound Route Filtering

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 27: Feature Information for BGP Prefix-Based Outbound Route Filtering

Feature Name	Releases	Feature Information
BGP Prefix-Based Outbound Route Filtering		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.



BGP Route-Map Continue Support for Outbound Policy

The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps.

- [Finding Feature Information](#), page 329
- [Information About BGP Route-Map Continue Support for Outbound Policy](#), page 330
- [How to Filter Traffic Using Continue Clauses in a BGP Route Map](#), page 332
- [Configuration Examples for BGP Route-Map Continue Support for Outbound Policy](#), page 335
- [Additional References](#), page 337
- [Feature Information for BGP Route-Map Continue Support for Outbound Policy](#), page 337

Finding Feature Information

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

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

Information About BGP Route-Map Continue Support for Outbound Policy

BGP Route Map with a Continue Clause

Subsequent to the Cisco implementation of route maps, the continue clause was introduced into BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering. The continue clause introduces the ability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow you 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.

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 (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.

**Note**

If the number of community lists in a match community clause within a route map exceed 256 characters in a line, you must nvgen multiple match community statements in a new line.

Set Operations with Continue Clauses

Set clauses are saved during the match clause evaluation process and are 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, 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 the “Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map” section.

How to Filter Traffic Using Continue Clauses in a BGP Route Map

Filtering Traffic Using Continue Clauses in a BGP Route Map

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. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
10. **match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*]
11. **set community** { { [*community-number*] [*well-known-community*] [**additive**] } | **none** }
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode, and creates a BGP routing process.

	Command or Action	Purpose
Step 4	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 10.0.0.1 remote-as 50000</pre>	<p>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 device.</p>
Step 5	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the device is placed in address family configuration mode for the IPv4 unicast address family if the unicast keyword is not specified. • The multicast keyword specifies IPv4 multicast address prefixes. • The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in</pre>	<p>Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
Step 9	<p>route-map <i>map-name</i> {permit deny}</p> <p>[<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map ROUTE-MAP-NAME permit 10</pre>	<p>Enters route-map configuration mode to create or configure a route map.</p>

	Command or Action	Purpose
Step 10	<p>match ip address {<i>access-list-number</i> <i>access-list-name</i>} [... <i>access-list-number</i> ... <i>access-list-name</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 1</pre>	<p>Configures a match command that specifies the conditions under which policy routing and route filtering occur.</p> <ul style="list-style-type: none"> 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. <p>Note The match and set commands used in this task are examples that are used to help describe the operation of the continue command. For a list of specific match and set commands, see the continue command in the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 11	<p>set community { {<i>community-number</i>} [<i>well-known-community</i>] [additive]} none}</p> <p>Example:</p> <pre>Device(config-route-map)# set community 10:1</pre>	<p>Configures a set command that specifies the routing action to perform if the criteria enforced by the match commands are met.</p> <ul style="list-style-type: none"> Multiple set commands can be configured. In this example, a clause is created to set the specified community number in aa:nn format.
Step 12	<p>continue [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# continue</pre>	<p>Configures a route map to continue to evaluate and execute match statements after a successful match occurs.</p> <ul style="list-style-type: none"> If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. 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 “implied continue.”
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	<p>Exits route-map configuration mode and enters privileged EXEC mode.</p>
Step 14	<p>show route-map [<i>map-name</i>]</p> <p>Example:</p> <pre>Device# show route-map</pre>	<p>(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.</p>

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.

```
Device# 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:
  Continue: 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 BGP Route-Map Continue Support for Outbound Policy

Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map

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 the number of successful match clauses, the as-path is prepended by one, two, or three autonomous system numbers.

```
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, 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** command. 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

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Route-Map Continue Support for Outbound Policy

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 [http://www.cisco.com/cisco/web/featurenavigator](#). An account on Cisco.com is not required.

Table 28: Feature Information for BGP Route-Map Continue Support for Outbound Policy

Feature Name	Releases	Feature Information
BGP Route-Map Continue Support for Outbound Policy	12.2(33)SXI	The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps.



Removing Private AS Numbers from the AS Path in BGP

Private autonomous system numbers (ASNs) are used by ISPs and customer networks to conserve globally unique AS numbers. Private AS numbers cannot be used to access the global Internet because they are not unique. AS numbers appear in eBGP AS paths in routing updates. Removing private ASNs from the AS path is necessary if you have been using private ASNs and you want to access the global Internet.

- [Finding Feature Information, page 339](#)
- [Restrictions on Removing and Replacing Private ASNs from the AS Path, page 339](#)
- [Information About Removing and Replacing Private ASNs from the AS Path, page 340](#)
- [How to Remove and Replace Private ASNs from the AS Path, page 341](#)
- [Configuration Examples for Removing and Replacing Private ASNs from the AS Path, page 345](#)
- [Additional References, page 348](#)
- [Feature Information for Removing and Replacing Private ASNs from the AS Path, page 349](#)

Finding Feature Information

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

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

Restrictions on Removing and Replacing Private ASNs from the AS Path

- The feature applies to eBGP neighbors only.

- The feature applies to routers in a public AS only. The workaround to this restriction would be to apply the **neighbor local-as** command on a per-neighbor basis, with the local AS number being a public AS number.

Information About Removing and Replacing Private ASNs from the AS Path

Public and Private AS Numbers

Public AS numbers are assigned by InterNIC and are globally unique. They range from 1 to 64511. Private AS numbers are used to conserve globally unique AS numbers, and they range from 64512 to 65535. Private AS numbers cannot be leaked to a global BGP routing table because they are not unique, and BGP best path calculations require unique AS numbers. Therefore, it might be necessary to remove private AS numbers from an AS path before the routes are propagated to a BGP peer.

Benefit of Removing and Replacing Private ASNs from the AS Path

External BGP requires that globally unique AS numbers be used when routing to the global Internet. Using private AS numbers (which are not unique) would prevent access to the global Internet. This feature allows routers that belong to a private AS to access the global Internet. A network administrator configures the routers to remove private AS numbers from the AS path contained in outgoing update messages and optionally, to replace those numbers with the ASN of the local router, so that the AS Path length remains unchanged.

Former Restrictions to Removing Private ASNs from the AS Path

The ability to remove private AS numbers from the AS path has been available for a long time. Prior to Cisco IOS Release 15.1(2)T, this feature had the following restrictions:

- If the AS path included both private and public AS numbers, using the **neighbor remove-private-as** command would not remove the private AS numbers.
- If the AS path contained confederation segments, using the **neighbor remove-private-as** command would remove private AS numbers only if the private AS numbers followed the confederation portion of the autonomous path.
- If the AS path contained the AS number of the eBGP neighbor, the private AS numbers would not be removed.

Enhancements to Removing Private ASNs from the AS Path

The ability to remove and replace private AS numbers from the AS path is enhanced in the following ways:

- The **neighbor remove-private-as** command will remove private AS numbers from the AS path even if the path contains both public and private ASNs.

- The **neighbor remove-private-as** command will remove private AS numbers even if the AS path contains only private AS numbers. There is no likelihood of a 0-length AS path because this command can be applied to eBGP peers only, in which case the AS number of the local router is appended to the AS path.
- The **neighbor remove-private-as** command will remove private AS numbers even if the private ASNs appear before the confederation segments in the AS path.
- The **replace-as** keyword is available to replace the private AS numbers being removed from the path with the local AS number, thereby retaining the same AS path length.
- The feature can be applied to neighbors per address family (address family configuration mode). Therefore, you can apply the feature for a neighbor in one address family and not on another, affecting update messages on the outbound side for only the address family for which the feature is configured.
- The feature can be applied in peer group template mode.
- When the feature is configured, output from the **show ip bgp update-group** and **show ip bgp neighbor** commands indicates that private AS numbers were removed or replaced.

How to Remove and Replace Private ASNs from the AS Path

Removing and Replacing Private ASNs from the AS Path (Cisco IOS Release 15.1(2)T and Later)

To remove private AS numbers from the AS path on the outbound side of an eBGP neighbor, perform the following task. To also replace private AS numbers with the local router's AS number, include the **all replace-as** keywords in Step 17.

The examples in this task reflect the configuration for Router 2 in the scenario in the figure below.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **network** *network-number*
14. **network** *network-number*
15. **neighbor** {*ip-address | ipv6-address[%] | peer-group-name*} **remote-as** *autonomous-system-number*
16. **neighbor** {*ip-address | ipv6-address[%] | peer-group-name*} **remote-as** *autonomous-system-number*
17. **neighbor** {*ip-address | peer-group-name*} **remove-private-as** [**all** [**replace-as**]]
18. **end**
19. **show ip bgp update-group**
20. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface gigabitethernet 0/0	Configures an interface.

	Command or Action	Purpose
Step 4	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 172.30.1.1 255.255.0.0</pre>	Sets a primary or secondary IP address for an interface.
Step 5	exit Example: <pre>Router(config-if)# exit</pre>	Returns to the next highest configuration mode.
Step 6	interface <i>type number</i> Example: <pre>Router(config)# interface serial 0/0</pre>	Configures an interface.
Step 7	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 172.16.0.2 255.255.255.0</pre>	Sets a primary or secondary IP address for an interface.
Step 8	exit Example: <pre>Router(config-if)# exit</pre>	Returns to the next highest configuration mode.
Step 9	interface <i>type number</i> Example: <pre>Router(config)# interface serial 1/0</pre>	Configures an interface.
Step 10	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 192.168.0.1 255.255.255.0</pre>	Sets a primary or secondary IP address for an interface.
Step 11	exit Example: <pre>Router(config-if)# exit</pre>	Returns to the next highest configuration mode.

	Command or Action	Purpose
Step 12	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 5</pre>	Specifies a BGP instance.
Step 13	<p>network <i>network-number</i></p> <p>Example:</p> <pre>Router(config-router)# network 172.30.0.0</pre>	Specifies a network to be advertised by BGP.
Step 14	<p>network <i>network-number</i></p> <p>Example:</p> <pre>Router(config-router)# network 192.168.0.0</pre>	Specifies a network to be advertised by BGP.
Step 15	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i>[%]} <i>peer-group-name</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 172.16.0.1 remote-as 65000</pre>	<p>Adds an entry to the routing table.</p> <ul style="list-style-type: none"> This example configures Router 3 as an eBGP neighbor in private AS 65000.
Step 16	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i>[%]} <i>peer-group-name</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.0.2 remote-as 1</pre>	<p>Adds an entry to the routing table.</p> <ul style="list-style-type: none"> This example configures Router 1 as an eBGP neighbor in public AS 1.
Step 17	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remove-private-as [all [replace-as]]</p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.0.2 remove-private-as all replace-as</pre>	<p>Removes private AS numbers from the AS Path in outgoing updates.</p> <ul style="list-style-type: none"> This example removes the private AS numbers from the AS path in outgoing eBGP updates and replaces them with 5, which is the public AS number of the local router.
Step 18	<p>end</p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Ends the current configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 19	show ip bgp update-group Example: Router# show ip bgp update-group	(Optional) Displays information about BGP update groups.
Step 20	show ip bgp neighbors Example: Router# show ip bgp neighbors	(Optional) Displays information about BGP neighbors.

Configuration Examples for Removing and Replacing Private ASNs from the AS Path

Example: Removing Private ASNs (Cisco IOS Release 15.1(2)T)

In the example below, Router A has the **neighbor remove-private-as** command configured, which removes private AS numbers in updates sent to the neighbor at 172.30.0.7. The subsequent **show** command asks for information about the route to host 1.1.1.1. The output includes private AS numbers 65200, 65201, 65201 in the AS path of 1001 65200 65201 65201 1002 1003 1003.

To prove that the private AS numbers were removed from the AS path, the **show** command on Router B also asks for information about the route to host 1.1.1.1. The output indicates a shorter AS path of 100 1001 1002 1003 1003, which excludes private AS numbers 65200, 65201, and 65201. The 100 prepended in the path is Router B's own AS number.

Router A

```
router bgp 100
  bgp log-neighbor-changes
  neighbor 19.0.101.1 remote-as 1001
  neighbor 172.30.0.7 remote-as 200
  neighbor 172.30.0.7 remove-private-as all
  no auto-summary

RouterA# show ip bgp 1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1          2
  1001 65200 65201 65201 1002 1003 1003
    19.0.101.1 from 19.0.101.1 (19.0.101.1)
      Origin IGP, localpref 100, valid, external, best RouterA#
```

Router B (All Private ASNs Have Been Removed)

```
RouterB# show ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 3
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  100 1001 1002 1003 1003
    172.30.0.6 from 172.30.0.6 (19.1.0.1)
      Origin IGP, localpref 100, valid, external, best RouterB#
```

Example: Removing and Replacing Private ASNs (Cisco IOS Release 15.1(2)T)

In the following example, when Router A sends prefixes to the peer 172.30.0.7, all private ASNs in the AS path are replaced with the router's own ASN, which is 100.

Router A

```
router bgp 100
  bgp log-neighbor-changes
  neighbor 172.16.101.1 remote-as 1001
  neighbor 172.16.101.1 update-source Loopback0
  neighbor 172.30.0.7 remote-as 200
  neighbor 172.30.0.7 remove-private-as all replace-as
  no auto-summary
```

Router A receives 1.1.1.1 from peer 172.16.101.1 which has some private ASNs (65200, 65201, and 65201) in the AS path list, as shown in the following output:

```
RouterA# show ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1          2
  1001 65200 65201 65201 1002 1003 1003
    172.16.101.1 from 172.16.101.1 (172.16.101.1)
      Origin IGP, localpref 100, valid, external, best RouterA#
```

Because Router A is configured with **neighbor 172.30.0.7 remove-private-as all replace-as**, Router A sends prefix 1.1.1.1 with all private ASNs replaced with 100:

Router B

```
RouterB# show ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 3
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  100 1001 100 100 100 1002 1003 1003
    172.30.0.6 from 172.30.0.6 (192.168.1.2)
      Origin IGP, localpref 100, valid, external, best RouterB#
```

Router B

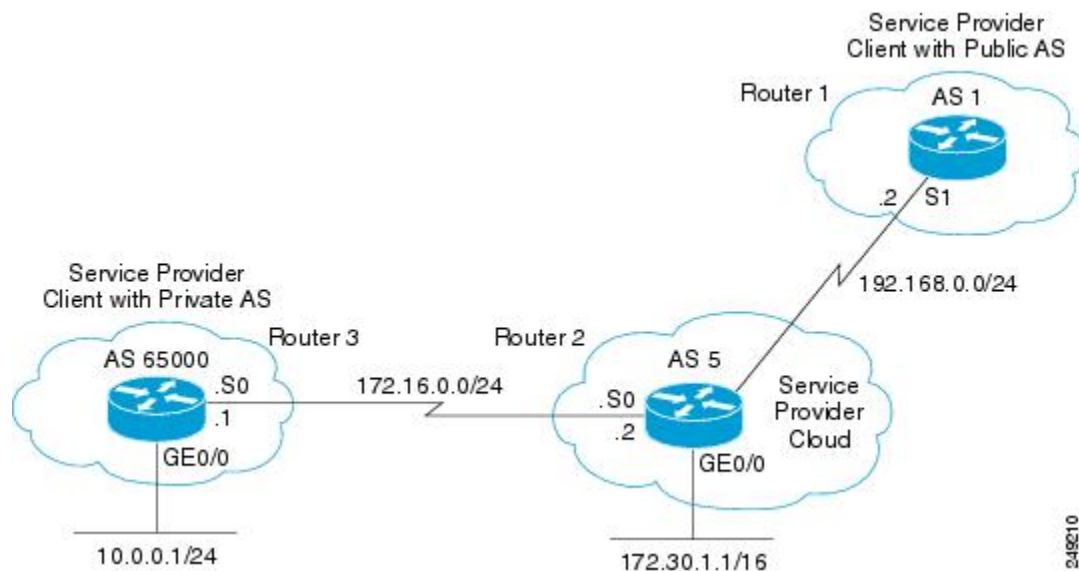
```
router bgp 200
  bgp log-neighbor-changes
  neighbor 172.30.0.6 remote-as 100
  no auto-summary
```

Example: Removing Private ASNs (Cisco IOS Release 12.2)

In this example, Router 3 uses private ASN 65000. Router 1 and Router 2 use public ASNs AS 1 and AS 5 respectively.

The figure below illustrates Router 2 belonging to a service provider, with Router 1 and Router 3 as its clients.

Figure 34: Removing Private AS Numbers



In this example, Router 2, belonging to the Service Provider, removes private AS numbers as follows.

- 1 Router 3 advertises the network 10.0.0.0/24 with the AS path attribute 65000 to Router 2.
- 2 Router 2 receives the update from Router 3 and makes an entry for the network 10.0.0.0/24 in its routing table with the next hop as 172.16.0.1 (serial interface S0 on Router 3).
- 3 Router 2 (service provider device), when configured with the **neighbor 192.168.0.2 remove-private-as** command, strips off the private AS number and constructs a new update packet with its own AS number as the AS path attribute for the 10.0.0.0/24 network and sends the packet to Router 1.
- 4 Router 1 receives the eBGP update for network 10.0.0.0/24 and makes an entry in its routing table with the next hop as 192.168.0.1 (serial interface S1 on Router 2). The AS path attribute for this network as seen on Router 1 is AS 5 (Router 2). Thus, the private AS numbers are prevented from entering the BGP tables of the Internet.

The configurations of Router 3, Router 2, and Router 1 follow.

Router 3

```
interface gigabitethernet 0/0
 ip address 10.0.0.1 255.255.255.0
!
interface Serial 0
 ip address 172.16.0.1 255.255.255.0
!
```

```

router bgp 65000
 network 10.0.0.0 mask 255.255.255.0
 neighbor 172.16.0.2 remote-as 5
!---Configures Router 2 as an eBGP neighbor in public AS 5.
!
end

```

Router 2

```

interface gigabitethernet 0/0
 ip address 172.30.1.1 255.255.0.0
!
interface Serial 0
 ip address 172.16.0.2 255.255.255.0
!
interface Serial 1
 ip address 192.168.0.1 255.255.255.0
!
router bgp 5
 network 172.30.0.0
 network 192.168.0.0
 neighbor 172.16.0.1 remote-as 65000
!---Configures Router 3 as an eBGP neighbor in private AS 65000.
 neighbor 192.168.0.2 remote-as 1
!---Configures Router 1 as an eBGP neighbor in public AS 1.
 neighbor 192.168.0.2 remove-private-as
!---Removes the private AS numbers from outgoing eBGP updates.
!
end

```

Router 1

```

version 12.2
!
!
interface Serial 0
 ip address 192.168.0.2 255.255.255.0
!
router bgp 1
 neighbor 192.168.0.1 remote-as 5
!---Configures Router 2 as an eBGP neighbor in public AS 5.
!
end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Removing and Replacing Private ASNs from the AS Path

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 29: Feature Information for BGP—Remove/Replace Private AS

Feature Name	Releases	Feature Information
BGP--Remove/Replace Private AS	15.1(2)T 15.0(1)S 15.1(1)SY	<p>Private autonomous system (AS) numbers are used by ISPs and customer networks to conserve globally unique AS numbers. Private AS numbers cannot be used to access the global Internet because they are not unique. AS numbers appear in eBGP AS paths in routing tables. Removing private AS numbers from the AS path is necessary if you have been using private AS numbers and you want to access the global Internet.</p> <p>The following command is modified:</p> <ul style="list-style-type: none"> • neighbor remove-private-as



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 designed to provide loop-free routing between organizations. This module contains tasks that use BGP neighbor session commands to configure:

- 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
- [Finding Feature Information, page 351](#)
- [Information About Configuring BGP Neighbor Session Options, page 352](#)
- [How to Configure BGP Neighbor Session Options, page 355](#)
- [Configuration Examples for BGP Neighbor Session Options, page 375](#)
- [Where to Go Next, page 378](#)
- [Additional References, page 378](#)
- [Feature Information for Configuring BGP Neighbor Session Options, page 380](#)

Finding Feature Information

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

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

Information About Configuring BGP Neighbor Session Options

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 uses the term “neighbor” over “peer.”

BGP Support for Fast Peering Session Deactivation

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco 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 devices 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 Release 12.4(4)T, 12.2(31)SB, 12.2(33)SRB, 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**

The **neighbor fall-over** command is not supported in Cisco IOS Release 15.0(1)SY. The **route-map** and *map-name* keyword-argument pair in the **bgp nexthop** command are not supported in Cisco IOS Release 15.0(1)SY.



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 Release 15.1(2)S 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.

TTL Security Check for BGP Neighbor Sessions

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

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 software, 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 configuration mode or address family configuration mode. For more details, see the "Disabling TCP Path MTU Discovery Globally for All BGP Sessions" section or the "Disabling TCP Path MTU Discovery for a Single BGP Neighbor" section.

How to Configure BGP Neighbor Session Options

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

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. Enter one of the following commands:
 - **address-family ipv4** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
 - **address-family ipv6** [**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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	Enter one of the following commands: <ul style="list-style-type: none"> • address-family ipv4 [unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] • address-family ipv6 [unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 unicast vrf blue	Enters address family configuration mode and enables IPv4 or IPv6 addressing. Perform this step when configuring fast session deactivation for a VRF address-family. Note Step 4 is only required if you are configuring fast session deactivation on a VRF. If you are not configuring fast session deactivation on a VRF, skip this step and perform the following commands under router BGP mode (config-router) rather than address family configuration mode (config-router-af).

	Command or Action	Purpose
Step 5	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 remote-as 50000</pre>	Establishes a peering session with a BGP neighbor.
Step 6	<p>neighbor <i>ip-address</i> fall-over</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 fall-over</pre>	<p>Configures the BGP peering to use fast session deactivation.</p> <ul style="list-style-type: none"> • BGP will remove all routes learned through this peer if the session is deactivated.
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

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.



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. **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 / length* | **permit** *network / 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	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 5	neighbor <i>ip-address</i> fall-over [route-map <i>map-name</i>]	Applies a route map when a route to the BGP changes. <ul style="list-style-type: none"> • In this example, the route map named CHECK-NBR is applied when the route to neighbor 192.168.1.2 changes.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR</pre>	
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 7	<p>ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] {deny <i>network / length</i> permit <i>network / length</i>} [ge <i>ge-value</i>] [le <i>le-value</i>]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28</pre>	<p>Creates a prefix list for BGP next-hop route filtering.</p> <ul style="list-style-type: none"> • 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.
Step 8	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map CHECK-NBR permit 10</pre>	<p>Configures a route map and enters route-map configuration mode.</p> <ul style="list-style-type: none"> • 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.
Step 9	<p>match ip address prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address prefix-list FILTER28</pre>	<p>Matches the IP addresses in the specified prefix list.</p> <ul style="list-style-type: none"> • Use the <i>prefix-list-name</i> argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and enters privileged EXEC mode.

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 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 BFD for BGP IPv6 Neighbors

In Cisco IOS Release 15.1(2)S 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

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Device(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	ipv6 cef Example: Device(config)# ipv6 cef	Enables Cisco Express Forwarding for IPv6.
Step 5	interface type number Example: Device(config)# interface fastethernet 0/1	Configures an interface type and number.
Step 6	ipv6 address ipv6-address / prefix-length Example: Device(config-if)# ipv6 address 2001:DB8:1:1::1/64	Configures an IPv6 address and enables IPv6 processing on an interface.
Step 7	bfd interval milliseconds min_rx milliseconds multiplier multiplier-value Example: Device(config-if)# bfd interval 500 min_rx 500 multiplier 3	Sets the baseline BFD session parameters on an interface.
Step 8	no shutdown Example: Device(config-if)# no shutdown	Restarts an interface.
Step 9	exit Example: Device(config-if)# exit	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 10	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 11	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	Disables the default IPv4 unicast address family for establishing peering sessions. • We recommend configuring this command in the global scope.
Step 12	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Device(config-router)# address-family ipv6	Enters address family configuration mode and enables IPv6 addressing.
Step 13	neighbor <i>ipv6-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router-af)# neighbor 2001:DB8:2:1::4 remote-as 45000	Adds the IP address of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table of the local router.
Step 14	neighbor <i>ipv6-address</i> fall-over bfd Example: Device(config-router-af)# neighbor 2001:DB8:2:1::4 fall-over bfd	Enables BGP to monitor the peering session of an IPv6 neighbor using BFD.
Step 15	end Example: Device(config-router-af)# end	Exits address family configuration mode and enters privileged EXEC mode.

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.

Before You Begin

- 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

- enable**
- trace** *[protocol] destination*
- configure terminal**
- router bgp** *autonomous-system-number*
- neighbor** *ip-address* **ttl-security hops** *hop-count*
- end**
- show running-config**
- show ip bgp neighbors** *[ip-address]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	trace <i>[protocol] destination</i> Example: Device# trace ip 10.1.1.1	Discovers the routes of the specified protocol that packets will actually take when traveling to their destination. <ul style="list-style-type: none"> Enter the trace command to determine the number of hops to the specified peer.

	Command or Action	Purpose
Step 3	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 4	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode, and creates a BGP routing process.
Step 5	<p>neighbor ip-address ttl-security hops hop-count</p> <p>Example:</p> <pre>Device(config-router)# neighbor 10.1.1.1 ttl-security hops 2</pre>	<p>Configures the maximum number of hops that separate two peers.</p> <ul style="list-style-type: none"> • The <i>hop-count</i> 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 <i>hop-count</i> 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	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 7	<p>show running-config</p> <p>Example:</p> <pre>Device# show running-config begin bgp</pre>	<p>(Optional) Displays the contents of the currently running configuration file.</p> <ul style="list-style-type: none"> • 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 8	<p>show ip bgp neighbors [ip-address]</p> <p>Example:</p> <pre>Device# show ip bgp neighbors 10.4.9.5</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> • This command displays "External BGP neighbor may be up to <i>number</i> hops away" when the BGP Support for TTL Security Check feature is

	Command or Action	Purpose
		<p>enabled. The <i>number</i> value represents the hop count. It is a number from 1 to 254.</p> <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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
  neighbor 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

      Sent      Rcvd
  Opens:          2         2
  Notifications:  0         0
  Updates:        0         0
  Keepalives:    226       227
  Route Refresh:  0         0
  Total:         228       229
  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

      Sent      Rcvd
```

```

Prefix activity:          ----      ----
  Prefixes Current:      0          0
  Prefixes Total:        0          0
  Implicit Withdraw:     0          0
  Explicit Withdraw:     0          0
  Used as bestpath:      n/a        0
  Used as multipath:     n/a        0
                               Outbound  Inbound
Local Policy Denied Prefixes:  -----  -----
  Total:                  0          0
Number of NLRI 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):
Timer           Starts    Wakeups      Next
Retrans         63         0            0x0
TimeWait        0          0            0x0
AckHold         62         50           0x0
SendWnd         0          0            0x0
KeepAlive       0          0            0x0
GiveUp          0          0            0x0
PmtuAger        0          0            0x0
DeadWait        0          0            0x0
iss: 712702676  snduna: 712703881  sndnxt: 712703881  sndwnd: 15180
irs: 2255946817 rcvnxt: 2255948041 rcvwnd: 15161  delrcvwnd: 1223
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
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

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.

Before You Begin

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ip bgp neighbors [ip-address] Example: Device# show ip bgp neighbors	(Optional) Displays information about the TCP and BGP connections to neighbors. <ul style="list-style-type: none"> • Use this command to determine whether BGP neighbors have TCP path MTU discovery enabled. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	router bgp autonomous-system-number Example: Device(config)# router bgp 50000	Enters router configuration mode to create or configure a BGP routing process.
Step 5	no bgp transport path-mtu-discovery Example: Device(config-router)# no bgp transport path-mtu-discovery	Disables TCP path MTU discovery for all BGP sessions.

	Command or Action	Purpose
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	show ip bgp neighbors [ip-address] Example: Device# show ip bgp neighbors	(Optional) Displays information about the TCP and BGP connections to neighbors. <ul style="list-style-type: none"> • In this example, the output from this command will not display that any neighbors have TCP path MTU enabled. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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
```

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 mode or address family configuration mode.

Before You Begin

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*] | *vpn4* [*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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	address-family {ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] vpv4 [unicast]} Example: Device(config-router)# address-family ipv4 unicast	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. <ul style="list-style-type: none"> The example creates an IPv4 unicast address family session.
Step 5	neighbor {ip-address peer-group-name} remote-as <i>autonomous-system-number</i> Example: Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000	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 6	neighbor {ip-address peer-group-name} activate Example: Device(config-router-af)# neighbor 172.16.1.1 activate	Activates the neighbor under the IPv4 address family.
Step 7	no neighbor {ip-address peer-group-name} transport {connection-mode path-mtu-discovery} Example: Device(config-router-af)# no neighbor 172.16.1.1 transport path-mtu-discovery	Disables TCP path MTU discovery for a single BGP neighbor. <ul style="list-style-type: none"> In this example, TCP path MTU discovery is disabled for the neighbor at 172.16.1.1.
Step 8	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	show ip bgp neighbors Example: Device# show ip bgp neighbors	(Optional) Displays information about the TCP and BGP connections to neighbors. <ul style="list-style-type: none"> In this example, the output from this command will not display that the neighbor has TCP path MTU discovery enabled. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Command or Action	Purpose
-------------------	---------

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

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.

Before You Begin

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	bgp transport path-mtu-discovery Example: Device(config-router)# bgp transport path-mtu-discovery	Enables TCP path MTU discovery for all BGP sessions.
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	show ip bgp neighbors Example: Device# show ip bgp neighbors	(Optional) Displays information about the TCP and BGP connections to neighbors. <ul style="list-style-type: none"> • In this example, the output from this command will show that all neighbors have TCP path MTU discovery enabled.

	Command or Action	Purpose
		Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i> .

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, KRRT: 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 mode 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*] | *vpn4* [*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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	address-family { ipv4 [mdt multicast unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn v4 [unicast]} Example: Device(config-router)# address-family ipv4 unicast	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. <ul style="list-style-type: none"> • The example creates an IPv4 unicast address family session.
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router-af)# neighbor 192.168.2.2 remote-as 50000	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 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.2 activate	Activates the neighbor under the IPv4 address family.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> } transport { connection-mode path-mtu-discovery } Example: Device(config-router-af)# neighbor 192.168.2.2 transport path-mtu-discovery	Enables TCP path MTU discovery for a single BGP neighbor.

	Command or Action	Purpose
Step 8	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	show ip bgp neighbors [<i>ip-address</i>] Example: Device# show ip bgp neighbors 192.168.2.2	(Optional) Displays information about the TCP and BGP connections to neighbors. Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i> .

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
.
.
.
  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
```

Configuration Examples for BGP Neighbor Session Options

Example: Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on device A and device B to monitor and use fast peering session deactivation for the neighbor session between the two devices. Although fast peering session deactivation is not required at both devices in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Device A

```
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
 neighbor 192.168.1.1 fall-over
 end
```

Device B

```
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:

```
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.

```
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: 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.

```
Router# trace ip 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.1.1
 0 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.

```
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

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.

```
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

The following example shows how to disable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2:

```
enable
configure terminal
router bgp 45000
neighbor 192.168.2.2 remote-as 50000
neighbor 192.168.2.2 activate
no neighbor 192.168.2.2 transport path-mtu-discovery
end
show ip bgp neighbors 192.168.2.2
```

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.

```
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.

```
enable
configure terminal
router bgp 45000
 neighbor 192.168.2.2 remote-as 50000
 neighbor 192.168.2.2 activate
 neighbor 192.168.2.2 transport path-mtu-discovery
end
show ip bgp neighbors 192.168.2.2
```

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” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	“Cisco BGP Overview” module
Conceptual and configuration details for basic BGP tasks	“Configuring a Basic BGP Network” module
Conceptual and configuration details for advanced BGP tasks	“Configuring Advanced BGP Features” module
Bidirectional Forwarding Detection configuration tasks	<i>IP Routing: BFD Configuration Guide</i>

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	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: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1191	<i>Path MTU Discovery</i>
RFC 1771	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>
RFC 1774	<i>BGP-4 Protocol Analysis</i>
RFC 1930	<i>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 30: Feature Information for Configuring BGP Neighbor Session Options Features

Feature Name	Releases	Feature Information
BGP Support for TCP Path MTU Discovery per Session	12.2(33)SRA 12.2(31)SB 12.2(33)SXH 12.4(20)T 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified by this feature: bgp transport, neighbor transport, show ip bgp neighbors.</p>

Feature Name	Releases	Feature Information
BGP Support for TTL Security Check	12.0(27)S 12.3(7)T 12.2(25)S 12.2(18)SXE 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified by this feature: neighbor ttl-security, show ip bgp neighbors.</p>
BGP IPv6 Client for Single-Hop BFD	15.1(2)S 15.2(3)T 15.2(4)S	<p>Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that use an IPv6 address.</p> <p>The following command was modified by this feature: neighbor fall-over.</p> <p>In Cisco IOS Release 15.2(4)S, support was added for the Cisco 7200 series router.</p>



BGP Neighbor Policy

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.

- [Finding Feature Information, page 383](#)
- [Information About BGP Neighbor Policy, page 383](#)
- [How to Display BGP Neighbor Policy Information, page 384](#)
- [Additional References, page 384](#)
- [Feature Information for BGP Neighbor Policy, page 385](#)

Finding Feature Information

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

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

Information About BGP Neighbor Policy

Benefit of BGP Neighbor Policy Feature

The BGP Neighbor Policy feature introduces new keywords to the **show ip bgp neighbors policy** command and the **show ip bgp template peer-policy** command 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.

How to Display BGP Neighbor Policy Information

Displaying BGP Neighbor Policy Information

SUMMARY STEPS

1. `enable`
2. `show ip bgp neighbors { ip-address | ipv6-address } policy [detail]`
3. `show ip bgp template peer-policy [policy-template-name [detail]]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>show ip bgp neighbors { ip-address ipv6-address } policy [detail]</code></p> <p>Example:</p> <pre>Device# show ip bgp neighbors 192.168.2.3 policy detail</pre>	<p>Displays the policies applied to the specified neighbor.</p>
Step 3	<p><code>show ip bgp template peer-policy [policy-template-name [detail]]</code></p> <p>Example:</p> <pre>Device# show ip bgp template peer-policy</pre>	<p>Displays the locally configured peer policy templates.</p>

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Neighbor Policy

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 31: Feature Information for BGP Neighbor Policy

Feature Name	Releases	Feature Information
BGP Neighbor Policy	15.1(1)SY	<p>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.</p> <p>The following commands were modified: show ip bgp neighbors, and show ip bgp template peer-policy.</p>



CHAPTER 15

BGP Dynamic Neighbors

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.

- [Finding Feature Information, page 387](#)
- [Information About BGP Dynamic Neighbors, page 387](#)
- [How to Configure BGP Dynamic Neighbors, page 388](#)
- [Configuration Examples for BGP Dynamic Neighbors, page 395](#)
- [Additional References, page 396](#)
- [Feature Information for BGP Dynamic Neighbors, page 397](#)

Finding Feature Information

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

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

Information About BGP Dynamic Neighbors

BGP Dynamic Neighbors

Support for the BGP Dynamic Neighbors feature was introduced in Cisco IOS Release 12.2(33)SXH on the Cisco Catalyst 6500 series switches. 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 Dynamic Neighbors

Implementing BGP Dynamic Neighbors Using Subnet Ranges

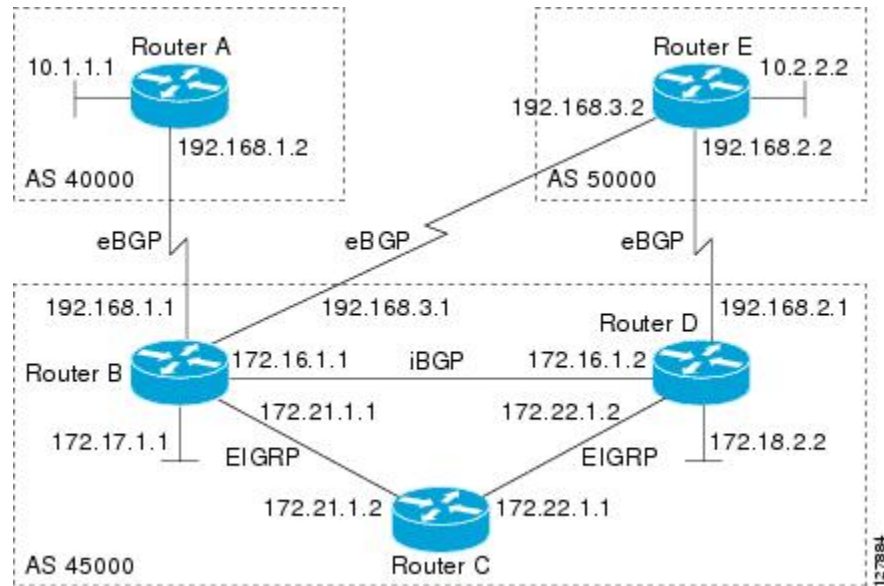
In Cisco IOS Release 12.2(33)SXH, support for BGP dynamic neighbors was introduced. Perform this task to implement the dynamic creation of BGP neighbors using subnet ranges.

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.

Figure 35: BGP Dynamic Neighbor Topology



Before You Begin

This task requires Cisco IOS Release 12.2(33)SXH, or a later release, to be running.



Note

This task supports only IPv4 BGP peering.

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** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
9. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
10. **address-family ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]]
11. **neighbor** {*ip-address* | *peer-group-name*} **activate**
12. **end**
13. Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.
14. **enable**
15. **configure terminal**
16. **router bgp** *autonomous-system-number*
17. **neighbor** {*ip-address*|*peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
18. Return to the first router.
19. **show ip bgp summary**
20. **show ip bgp peer-group** [*peer-group-name*] [**summary**]
21. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: DeviceB> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. • The configuration is entered on router B.
Step 2	configure terminal Example: DeviceB# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>DeviceB(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 4	<p>bgp log-neighbor-changes</p> <p>Example:</p> <pre>DeviceB(config-router)# bgp log-neighbor-changes</pre>	<p>(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets.</p> <ul style="list-style-type: none"> 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.
Step 5	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>DeviceB(config-router)# neighbor group192 peer-group</pre>	<p>Creates a BGP peer group.</p> <ul style="list-style-type: none"> In this example, a peer group named group192 is created. This group will be used as a listen range group.
Step 6	<p>bgp listen [limit <i>max-number</i>]</p> <p>Example:</p> <pre>DeviceB(config-router)# bgp listen limit 200</pre>	<p>Sets a global limit of BGP dynamic subnet range neighbors.</p> <ul style="list-style-type: none"> Use the optional limit keyword and <i>max-number</i> argument to define the maximum number of BGP dynamic subnet range neighbors that can be created. <p>Note Only the syntax applicable to this task is used in this example. For the complete syntax, see Step 7.</p>
Step 7	<p>bgp listen [limit <i>max-number</i> range <i>network / length</i> peer-group <i>peer-group-name</i>]</p> <p>Example:</p> <pre>DeviceB(config-router)# bgp listen range 192.168.0.0/16 peer-group group192</pre>	<p>Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature.</p> <ul style="list-style-type: none"> Use the optional limit keyword and <i>max-number</i> argument to define the maximum number of BGP dynamic neighbors that can be created. Use the optional range keyword and <i>network / length</i> argument to define a prefix range to be associated with the specified peer group. In this example, the prefix range 192.168.0.0/16 is associated with the listen range group named group192.
Step 8	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} ebgp-multihop [<i>tll</i>]</p> <p>Example:</p> <pre>DeviceB(config-router)# neighbor group192 ebgp-multihop 255</pre>	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
Step 9	<p>neighbor <i>peer-group-name</i> remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number...</i>]</p> <p>Example:</p> <pre>DeviceB(config-router)# neighbor group192 remote-as 40000 alternate-as 50000</pre>	<p>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.</p> <ul style="list-style-type: none"> • Use the optional alternate-as keyword and <i>autonomous-system-number</i> 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. <p>Note The alternate-as keyword is used only with the listen range peer groups, not with individual BGP neighbors.</p>
Step 10	<p>address-family ipv4 [mdt multicast unicast [vrf <i>vrf-name</i>]]</p> <p>Example:</p> <pre>DeviceB(config-router)# address-family ipv4 unicast</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p>
Step 11	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>DeviceB(config-router-af)# neighbor group192 activate</pre>	<p>Activates the neighbor or listen range peer group for the configured address family.</p> <ul style="list-style-type: none"> • In this example, the neighbor 172.16.1.1 is activated for the IPv4 address family. <p>Note Usually BGP peer groups cannot be activated using this command, but the listen range peer groups are a special case.</p>
Step 12	<p>end</p> <p>Example:</p> <pre>DeviceB(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
Step 13	<p>Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.</p>	—
Step 14	<p>enable</p> <p>Example:</p> <pre>DeviceE> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted. • The configuration is entered on Router E.
Step 15	<p>configure terminal</p> <p>Example:</p> <pre>DeviceE# configure terminal</pre>	<p>Enters global configuration mode.</p>

	Command or Action	Purpose
Step 16	router bgp <i>autonomous-system-number</i> Example: DeviceE(config)# router bgp 50000	Enters router configuration mode for the specified routing process.
Step 17	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number...</i>] Example: DeviceE(config-router)# neighbor 192.168.3.1 remote-as 45000	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. <ul style="list-style-type: none"> 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.
Step 18	Return to the first router.	—
Step 19	show ip bgp summary Example: DeviceB# show ip bgp summary	(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors. <ul style="list-style-type: none"> In this step, the configuration has returned to Router B.
Step 20	show ip bgp peer-group [<i>peer-group-name</i>] [summary] Example: DeviceB# show ip bgp peer-group group192	(Optional) Displays information about BGP peer groups.
Step 21	show ip bgp neighbors [<i>ip-address</i>] Example: DeviceB# show ip bgp neighbors 192.168.3.2	(Optional) Displays information about BGP and TCP connections to neighbors. <ul style="list-style-type: none"> 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. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

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 00:00:25, hold time is 180, keepalive intervals
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          Rcvd
Opens:           1           1
Notifications:   0           0
Updates:          0           0
Keepalives:       7           7
Route Refresh:    0           0
Total:            8           8
Default minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor version 1/0
Output queue size : 0
Index 1, Offset 0, Mask 0x2
1 update-group member
group192 peer-group member
.
.
.
```

Configuration Examples for BGP Dynamic Neighbors

Example: Implementing BGP Dynamic Neighbors Using Subnet Ranges

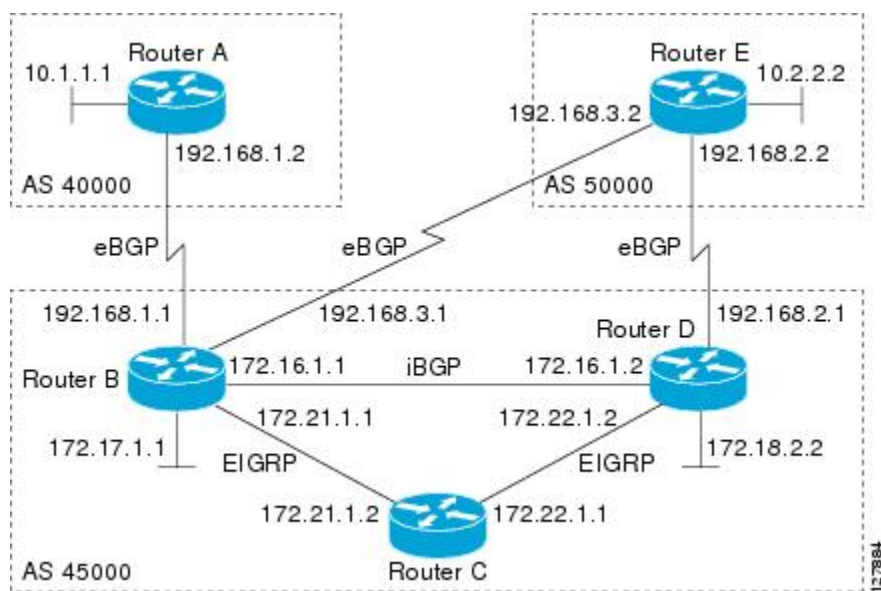
In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added 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 36: BGP Dynamic Neighbor Topology



Router B

```
enable
configure terminal
router bgp 45000
  bgp log-neighbor-changes
  bgp listen limit 200
  bgp listen range 172.21.0.0/16 peer-group group172
```

```

bgp listen range 192.168.0.0/16 peer-group group192
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
configure terminal
router bgp 40000
neighbor 192.168.1.1 remote-as 45000
exit

```

Router E

```

enable
configure terminal
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

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Dynamic Neighbors

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 32: Feature Information for BGP Dynamic Neighbors

Feature Name	Releases	Feature Information
BGP Dynamic Neighbors	12.2(33)SXH	<p>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.</p> <p>The following commands were introduced or modified by this feature: bgp listen, debug ip bgp range, neighbor remote-as, show ip bgp neighbors, show ip bgp peer-group, and show ip bgp summary.</p>



CHAPTER 16

BGP Support for Next-Hop Address Tracking

The BGP Support for 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 bestpath calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

- [Finding Feature Information, page 399](#)
- [Information About BGP Support for Next-Hop Address Tracking, page 399](#)
- [How to Configure BGP Support for Next-Hop Address Tracking, page 401](#)
- [Configuration Examples for BGP Support for Next-Hop Address Tracking, page 412](#)
- [Additional References, page 413](#)
- [Feature Information for BGP Support for Next-Hop Address Tracking, page 414](#)

Finding Feature Information

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

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

Information About BGP Support for Next-Hop Address Tracking

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.

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 device 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 device 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.

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 nexthop** 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

Use route map on ASR series devices to set the next hop as BGP peer for the route and apply that route map in outbound direction towards the peer.



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 Support for Fast Peering Session Deactivation

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco 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 devices 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.

How to Configure BGP Support for Next-Hop Address Tracking

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 “Configuring BGP Route Dampening.”

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. Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

For more examples of how to use the **bgp nexthop** command, see the “Examples: Configuring BGP Selective Next-Hop Route Filtering” section in this module.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>	Specifies the IPv4 address family and enters address family configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<ul style="list-style-type: none"> 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	<p>bgp nexthop route-map map-name</p> <p>Example:</p> <pre>Device(config-router-af)# bgp nexthop route-map CHECK-NEXTHOP</pre>	<p>Permits a route map to selectively define routes to help resolve the BGP next hop.</p> <ul style="list-style-type: none"> In this example the route map named CHECK-NEXTHOP is created.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
Step 8	<p>ip prefix-list list-name [seq seq-value] {deny network / length permit network/length} [ge ge-value] [le le-value]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25</pre>	<p>Creates a prefix list for BGP next-hop route filtering.</p> <ul style="list-style-type: none"> 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	<p>route-map map-name [permit deny] [sequence-number]</p> <p>Example:</p> <pre>Device(config)# route-map CHECK-NEXTHOP deny 10</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> 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.
Step 10	<p>match ip address prefix-list prefix-list-name [prefix-list-name...]</p>	<p>Matches the IP addresses in the specified prefix list.</p> <ul style="list-style-type: none"> Use the <i>prefix-list-name</i> argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-route-map)# match ip address prefix-list FILTER25</pre>	<p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 11	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 12	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map CHECK-NEXTHOP permit 20</pre>	<p>Configures a route map and enters route map configuration mode.</p> <ul style="list-style-type: none"> In this example, all other IP addresses are permitted by route map CHECK-NEXTHOP.
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	Exits route map configuration mode and enters privileged EXEC mode.
Step 14	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>]</p> <p>Example:</p> <pre>Device# show ip bgp</pre>	<p>Displays the entries in the BGP routing table.</p> <ul style="list-style-type: none"> Enter this command to view the next-hop addresses for each route. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Example

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
   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
*> 172.16.1.0/24  0.0.0.0          0           32768 i
*> 172.17.1.0/24  0.0.0.0          0           32768
```

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 64512	Enters router configuration mode to create or configure a BGP routing process.
Step 4	address-family ipv4 <i>[[mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] vpnv4 [unicast]]</i>	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	
Step 5	<p>bgp nexthop trigger delay <i>delay-timer</i></p> <p>Example:</p> <pre>Device(config-router-af)# bgp nexthop trigger delay 20</pre>	<p>Configures the delay interval between routing table walks for next-hop address tracking.</p> <ul style="list-style-type: none"> • The time period determines how long BGP will wait before starting a full routing table walk after notification is received. • The value for the <i>delay-timer</i> argument is a number from 1 to 100 seconds. The default value is 5 seconds. • The example configures a delay interval of 20 seconds.
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>Exits address-family configuration mode, and enters privileged EXEC mode.</p>

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. Beginning with Cisco IOS Release 12.2(33)SB6, BGP next-hop address tracking is also enabled by default under the VPNv6 address family whenever the next hop is an IPv4 address mapped to an IPv6 next-hop address.

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 reenale BGP next-hop address tracking, use the **bgp nexthop** 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**] | **vpn6** [**unicast**]]
5. **no bgp nexthop trigger enable**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 64512	Enters router configuration mod to create or configure a BGP routing process.
Step 4	address-family ipv4 [[mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast] vpn6 [unicast]] Example: Device(config-router)# address-family ipv4 unicast	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. <ul style="list-style-type: none"> • The example creates an IPv4 unicast address family session.
Step 5	no bgp nexthop trigger enable Example: Device(config-router-af)# no bgp nexthop trigger enable	Disables BGP next-hop address tracking. <ul style="list-style-type: none"> • Next-hop address tracking is enabled by default for IPv4 and VPNv4 address family sessions. • The example disables next-hop address tracking.
Step 6	end Example: Device(config-router-af)# end	Exits address-family configuration mode, and enters Privileged EXEC mode.

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

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Device(config-router)# address-family ipv4 unicast	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.

	Command or Action	Purpose
Step 5	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: <pre>Device(config-router-af)# neighbor 10.0.0.1 remote-as 50000</pre>	Establishes a peering session with a BGP neighbor.
Step 6	neighbor <i>ip-address</i> fall-over Example: <pre>Device(config-router-af)# neighbor 10.0.0.1 fall-over</pre>	Configures the BGP peering to use fast session deactivation. <ul style="list-style-type: none"> • BGP will remove all routes learned through this peer if the session is deactivated.
Step 7	end Example: <pre>Device(config-router-af)# end</pre>	Exits configuration mode and returns to privileged EXEC mode.

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.



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. **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 / length* | **permit** *network / 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	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 5	neighbor <i>ip-address</i> fall-over [route-map <i>map-name</i>]	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.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR</pre>	
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# exit</pre>	Exits router configuration mode and enters global configuration mode.
Step 7	<p>ip prefix-list list-name [seq seq-value]{deny network / length permit network / length}[ge ge-value] [le le-value]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28</pre>	<p>Creates a prefix list for BGP next-hop route filtering.</p> <ul style="list-style-type: none"> • 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.
Step 8	<p>route-map map-name [permit deny][sequence-number]</p> <p>Example:</p> <pre>Device(config)# route-map CHECK-NBR permit 10</pre>	<p>Configures a route map and enters route-map configuration mode.</p> <ul style="list-style-type: none"> • 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.
Step 9	<p>match ip address prefix-list prefix-list-name [prefix-list-name...]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address prefix-list FILTER28</pre>	<p>Matches the IP addresses in the specified prefix list.</p> <ul style="list-style-type: none"> • Use the <i>prefix-list-name</i> argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	Exits configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP Support for Next-Hop Address Tracking

Example: Enabling and Disabling BGP Next-Hop Address Tracking

In the following example, next-hop address tracking is disabled under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
  no bgp nexthop trigger enable
```

Example: Adjusting the Delay Interval for BGP Next-Hop Address Tracking

In the following example, the delay interval for next-hop tracking is configured to occur every 20 seconds under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
  bgp nexthop trigger delay 20
```

Examples: Configuring BGP Selective Next-Hop Route Filtering

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.

```
router bgp 45000
 address-family ipv4 unicast
  bgp nexthop route-map CHECK-BGP
  exit
  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.

```
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
```

Example: Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on device A and device B to monitor and use fast peering session deactivation for the neighbor session between the two devices. Although fast peering session deactivation is not required at both devices in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Device A

```
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
 neighbor 192.168.1.1 fall-over
end
```

Device B

```
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:

```
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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for Next-Hop Address Tracking

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 . An account on Cisco.com is not required.

Table 33: Feature Information for BGP Support for Next-Hop Address Tracking

Feature Name	Releases	Feature Information
BGP Support for Next-Hop Address Tracking	12.2(33)SXH 15.0(1)SY	<p>The BGP Support for Next-Hop Address Tracking feature is enabled by default when a supporting Cisco IOS 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.</p> <p>The following command was introduced in this feature: bgp nexthop.</p>
BGP Selective Address Tracking	15.0(1)SY 15.2(1)SY	<p>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.</p> <p>The following commands were modified by this feature: bgp nexthop, neighbor fall-over.</p>

Feature Name	Releases	Feature Information
BGP Support for Fast Peering Session Deactivation	12.2(33)SXH	<p>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.</p> <p>The following command was modified by this feature: neighbor fall-over.</p>



BGP Restart Neighbor Session After Max-Prefix Limit Reached

The BGP Restart Session After Max-Prefix Limit Reached feature adds the **restart** keyword to the **neighbor maximum-prefix** command. This allows a network operator to configure the time interval at which a peering session is reestablished by a device when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit.

- [Finding Feature Information, page 417](#)
- [Information About BGP Neighbor Session Restart After Max-Prefix Limit Reached, page 418](#)
- [How to Configure a Device to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded, page 419](#)
- [Configuration Example for BGP Restart Neighbor Session After Max-Prefix Limit Reached, page 422](#)
- [Additional References for BGP Restart Neighbor Session After Max-Prefix Limit Reached, page 422](#)
- [Feature Information for BGP Restart Neighbor Session after Max-Prefix Limit, page 423](#)

Finding Feature Information

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

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

Information About BGP Neighbor Session Restart After Max-Prefix Limit Reached

Prefix Limits and BGP Peering Sessions

Use the **neighbor maximum-prefix** command to limit the maximum number of prefixes that a device running BGP can receive from a peer. When the device receives too many prefixes from a peer 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, which clears stored prefixes.

BGP Neighbor Session Restart with the Maximum Prefix Limit

The **restart** keyword was added to the **neighbor maximum-prefix** command so that a network operator can configure a device to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down. The time interval at which peering can be reestablished automatically is configurable. The *restart-interval* for the **restart** keyword is specified in minutes; range is from 1 to 65,535 minutes.

Subcodes for BGP Cease Notification

Border Gateway Protocol (BGP) imposes maximum limits on the maximum number of prefixes that are accepted from a peer for a given address family. This limitation safeguards the device from resource depletion caused by misconfiguration, either locally or on the remote neighbor. To prevent a peer from flooding BGP with advertisements, a limit is placed on the number of prefixes that are accepted from a peer for each supported address family. The default limits can be overridden through configuration of the maximum-prefix limit command for the peer for the appropriate address family.

The following subcodes are supported for the BGP cease notification message:

- Maximum number of prefixes reached
- Administrative shutdown
- Peer de-configured
- Administrative reset

A cease notification message is sent to the neighbor and the peering with the neighbor is terminated when the number of prefixes received from the peer for a given address family exceeds the maximum limit (either set by default or configured by the user) for that address family. It is possible that the maximum number of prefixes for a neighbor for a given address family has been configured after the peering with the neighbor has been established and a certain number of prefixes have already been received from the neighbor for that address family. A cease notification message is sent to the neighbor and peering with the neighbor is terminated immediately after the configuration if the configured maximum number of prefixes is fewer than the number of prefixes that have already been received from the neighbor for the address family.

How to Configure a Device to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded

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 device 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 device 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 101</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} maximum-prefix <i>maximum</i> [<i>threshold</i>] [restart <i>minutes</i>] [warning-only]</p> <p>Example:</p> <pre>Device(config-router)# neighbor 10.4.9.5 maximum-prefix 1000 90 restart 60</pre>	<p>Configures the maximum-prefix limit on a router that is running BGP.</p> <ul style="list-style-type: none"> Use the restart keyword and <i>minutes</i> 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 <i>minutes</i> is from 1 to 65535 minutes. Use the warning-only keyword to configure the device to disable the restart capability to allow you to adjust a peer that is sending too many prefixes. <p>Note If the <i>minutes</i> argument is not configured, the disabled session will stay down after the maximum-prefix limit is exceeded. This is the default behavior.</p>
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.
Step 6	<p>show ip bgp neighbors <i>ip-address</i></p> <p>Example:</p> <pre>Device# show ip bgp neighbors 10.4.9.5</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> In this example, the output from this command will display the maximum prefix limit for the specified neighbor and the configured restart timer value. <p>Note Only the syntax applicable to this task is used in this example. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Examples

The following sample output from the **show ip bgp neighbors** command verifies that a device 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.

Device# **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
  BGP state = Established, up for 2w2d
  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      Rcvd
  Opens:          1          1
  Notifications:  0          0
  Updates:        0          0
  Keepalives:    23095     23095
  Route Refresh:  0          0
  Total:         23096     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 0x4
  Member of update-group 2

      Sent      Rcvd
  Prefix activity:
  Prefixes Current:    0          0
  Prefixes Total:     0          0
  Implicit Withdraw:  0          0
  Explicit Withdraw:  0          0
  Used as bestpath:   n/a         0
  Used as multipath:  n/a         0
                        Outbound   Inbound
  Local Policy Denied Prefixes:
  Total:                0          0
!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):
Timer      Starts      Wakeups      Next
Retrans    23098        0            0x0
TimeWait   0            0            0x0
AckHold    23096       22692        0x0
SendWnd    0            0            0x0
KeepAlive  0            0            0x0
GiveUp     0            0            0x0
PmtuAger   0            0            0x0
DeadWait   0            0            0x0
iss: 1900546793  snduna: 1900985663  sndnxt: 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 device that is running BGP from exceeding the maximum-prefix limit.

Display of the following error messages can indicate an underlying problem that is causing the neighbor session to become disabled. You should check the values configured for the **neighbor maximum-prefix** command 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 device that is sending an excessive number of prefixes. For more details about BGP route dampening, see the “Configuring Advanced BGP Features” module.

Configuration Example for BGP Restart Neighbor Session After Max-Prefix Limit Reached

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 device to reestablish a peering session after 30 minutes if one has been disabled:

```
router bgp 101
 network 172.16.0.0
 neighbor 192.168.6.6 maximum-prefix 2000 restart 30
```

Additional References for BGP Restart Neighbor Session After Max-Prefix Limit Reached

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 4486	<i>Subcodes for BGP Cease Notification Message</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Restart Neighbor Session after Max-Prefix Limit

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 [http://www.cisco.com/cisco/web/featurenavigator](#). An account on Cisco.com is not required.

Table 34: Feature Information for BGP Restart Session After Max-Prefix Limit

Feature Name	Releases	Feature Information
BGP Restart Session After Max-Prefix Limit		<p>The BGP Restart Session After Max-Prefix Limit Reached feature adds the restart keyword to the neighbor maximum-prefix command. This allows a network operator to configure the time interval at which a peering session is reestablished by a device when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit.</p> <p>The following commands were modified: neighbor maximum-prefix and show ip bgp neighbors.</p>
BGP—Subcodes for BGP Cease Notification		Support for subcodes for BGP cease notification has been added.



BGP Support for Dual AS Configuration for Network AS Migrations

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.

- [Finding Feature Information, page 425](#)
- [Information About BGP Support for Dual AS Configuration for Network AS Migrations, page 426](#)
- [How to Configure BGP Support for Dual AS Configuration for Network AS Migrations, page 427](#)
- [Configuration Examples for Dual-AS Peering for Network Migration, page 430](#)
- [Additional References, page 431](#)
- [Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations, page 432](#)

Finding Feature Information

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

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

Information About BGP Support for Dual AS Configuration for Network AS Migrations

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 Release 12.0(29)S, 12.3(14)T, 12.2(33)SXH, 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 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.

**Caution**

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.

BGP Network Migration to 4-Byte Autonomous System Numbers

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 started to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.

The Cisco implementation of 4-byte autonomous system numbers 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.

Migrating your BGP network to 4-byte autonomous system numbers requires some planning. 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.

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.

For details about steps to perform to upgrade a BGP network to full 4-byte autonomous system support, see the [Migration Guide for Explaining 4-Byte Autonomous System](#) white paper.

How to Configure BGP Support for Dual AS Configuration for Network AS Migrations

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** *regex* | **dampened-routes** | **received** *prefix-filter*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 40000	Enters router configuration mode, and creates a BGP routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i>	Establishes a peering session with a BGP neighbor.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 remote-as 45000</pre>	
Step 5	<p>neighbor <i>ip-address</i> local-as [<i>autonomous-system-number</i> [no-prepend [replace-as [dual-as]]]]</p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 local-as 50000 no-prepend replace-as dual-as</pre>	<p>Customizes the AS_PATH attribute for routes received from an eBGP neighbor.</p> <ul style="list-style-type: none"> The replace-as keyword is used to prepend only the local autonomous system number (as configured with the <i>ip-address</i> argument) to the AS_PATH attribute. The autonomous system number from the local BGP routing process is not prepended. 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 <i>ip-address</i> argument (<i>local-as</i>). The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the <i>local-as</i> number.
Step 6	<p>neighbor <i>ip-address</i> remove-private-as</p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 remove-private-as</pre>	<p>(Optional) Removes private autonomous system numbers from outbound routing updates.</p> <ul style="list-style-type: none"> 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. Private autonomous system numbers (64512 to 65535) are automatically removed from the AS_PATH attribute when this command is configured.
Step 7	<p>end</p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>
Step 8	<p>show ip bgp [<i>network</i>] [<i>network-mask</i>] [longer-prefixes] [prefix-list <i>prefix-list-name</i> route-map <i>route-map-name</i>] [shorter-prefixes <i>mask-length</i>]</p> <p>Example:</p> <pre>Router# show ip bgp</pre>	<p>Displays entries in the BGP routing table.</p> <ul style="list-style-type: none"> The output can be used to verify if the real autonomous system number or <i>local-as</i> number is configured.

	Command or Action	Purpose
Step 9	<pre>show ip bgp neighbors [neighbor-address] [received-routes routes advertised-routes paths regexp dampened-routes received prefix-filter]</pre> <p>Example:</p> <pre>Router# show ip bgp neighbors</pre>	<p>Displays information about TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> The output will display local AS, no-prepend, replace-as, and dual-as with the corresponding autonomous system number when these options are configured.

Configuration Examples for Dual-AS Peering for Network Migration

Example: Dual AS Configuration

The following examples shows 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)

```
interface Serial3/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 40000
 no synchronization
 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
```

Router 1 in Autonomous System 45000 (Provider Network)

```
interface 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)

```
interface 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:

```

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.

```

interface 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 an AS with Another AS in Routing Updates

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:

```

router bgp 64512
 neighbor 10.3.3.33 local-as 50000 no-prepend replace-as

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 35: Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations

Feature Name	Releases	Feature Information
BGP Support for Dual AS Configuration for Network AS Migrations	12.2(33)SXH	<p>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.</p> <p>The following command was modified by this feature: neighbor local-as.</p>



Configuring Internal BGP Features

This module describes how to configure internal Border Gateway Protocol (BGP) features. Internal BGP (iBGP) refers to running BGP on networking devices within one autonomous system. BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains (autonomous systems) that contain independent routing policies. Many companies now have large internal networks, and there are many issues involved in scaling the existing internal routing protocols to match the increasing traffic demands while maintaining network efficiency.

- [Finding Feature Information, page 433](#)
- [Information About Internal BGP Features, page 433](#)
- [How to Configure Internal BGP Features, page 439](#)
- [Configuration Examples for Internal BGP Features, page 453](#)
- [Additional References for Internal BGP Features, page 456](#)
- [Feature Information for Configuring Internal BGP Features, page 458](#)

Finding Feature Information

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

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

Information About Internal BGP Features

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 are preserved. This feature allows you to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

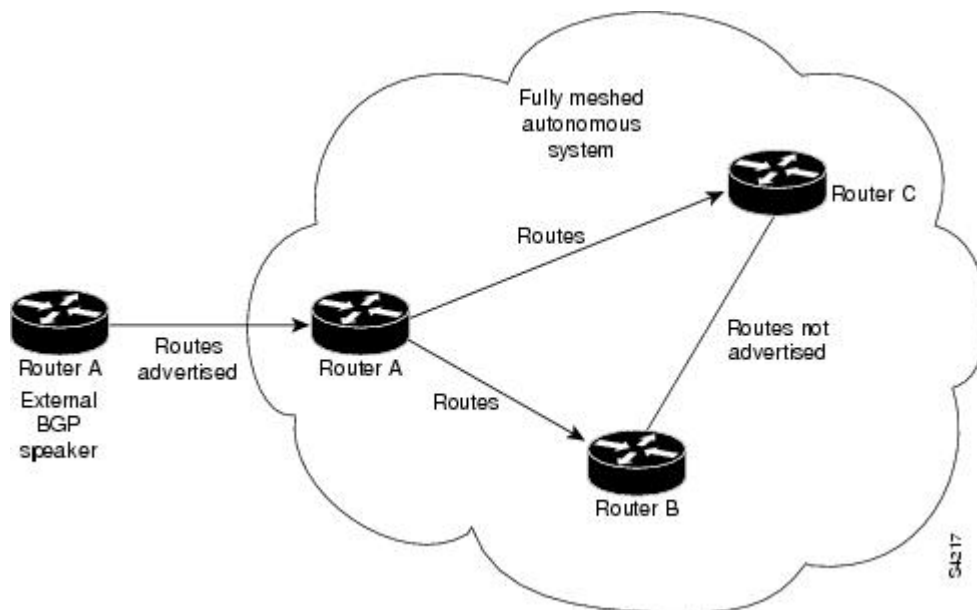
To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number.

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.

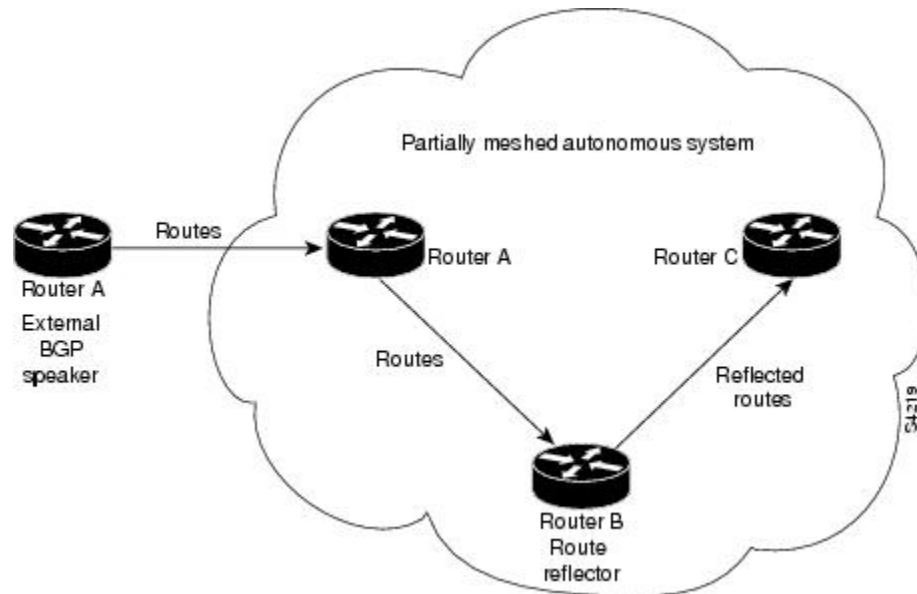
Figure 37: Three Fully Meshed iBGP Speakers



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 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.

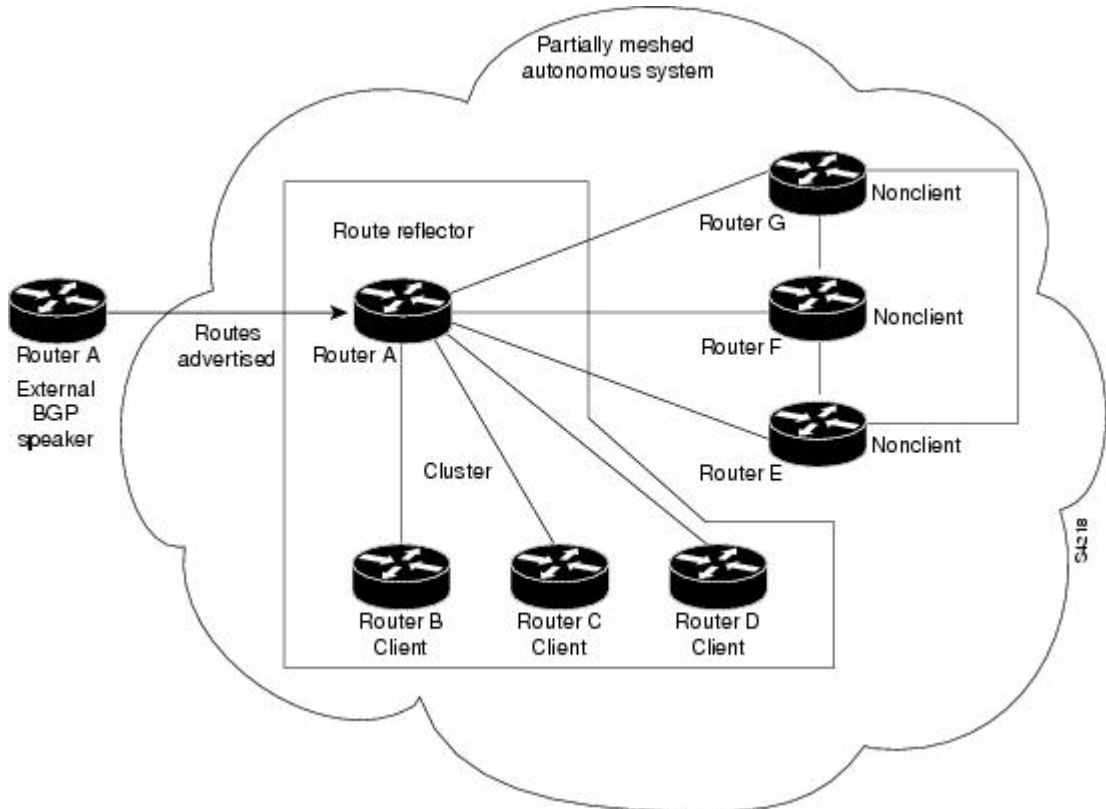
Figure 38: Simple BGP Model with a Route Reflector



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 39: 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

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 be attempted only by someone who has a good understanding of the design implications.

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

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.

BGP Route Map Next Hop Self

The BGP Route Map Next Hop Self feature provides a way to override the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath` selectively. These settings are global for an address family. For some routes this may not be appropriate. For example, static routes may need to be redistributed with a next hop of self, but connected routes and routes learned via Interior Border Gateway Protocol (iBGP) or Exterior Border Gateway Protocol (eBGP) may continue to be redistributed with an unchanged next hop.

The BGP route map next hop self functionality modifies the existing route map infrastructure to configure a new `ip next-hop self` setting, which overrides the `bgp next-hop unchanged` and `bgp next-hop unchanged allpaths` settings.

The `ip next-hop self` setting is applicable only to VPNv4 and VPNv6 address families. Routes distributed by protocols other than BGP are not affected.

You configure a new `bgp route-map priority` setting to inform BGP that the route map will take priority over the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath`. The `bgp route-map priority` setting only impacts BGP. The `bgp route-map priority` setting has no impact unless you configure the `bgp next-hop unchanged` or `bgp next-hop unchanged allpaths` settings.

How to Configure Internal BGP Features

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 <i>as-number</i>	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 <i>as-number</i> [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, on page 440.](#)"

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 { <i>ip-address</i> <i>peer-group-name</i> } 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 <i>cluster-id</i>	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 a Set Next Hop for an 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.

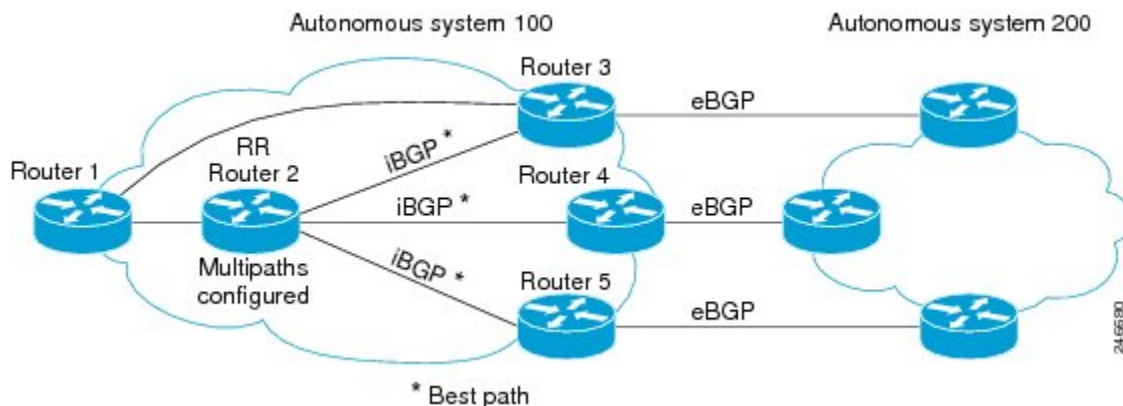


Note

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 40: Route Reflector Using a Route Map to a 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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> Example: Router(config)# route-map rr-out	Enters route map configuration mode to configure a route map. • The route map is created to set the next hop for the route reflector client.
Step 4	set ip next-hop <i>ip-address</i> Example: Router(config-route-map)# set ip next-hop 10.2.0.1	Specifies that for routes that are advertised where this route map is applied, the next-hop attribute is set to this IPv4 address. • For this task, we want to set the next hop to be the address of the RR.

	Command or Action	Purpose
Step 5	exit Example: <pre>Router(config-route-map)# exit</pre>	Exits route-map configuration mode and enters global configuration mode.
Step 6	router bgp <i>as-number</i> Example: <pre>Router(config)# router bgp 100</pre>	Enters router configuration mode and creates a BGP routing process.
Step 7	address-family ipv4 Example: <pre>Router(config-router-af)# address-family ipv4</pre>	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.
Step 8	maximum-paths ibgp <i>number</i> Example: <pre>Router(config-router)# maximum-paths ibgp 5</pre>	Controls the maximum number of parallel iBGP routes that can be installed in the routing table.
Step 9	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>Router(config-router-af)# neighbor 10.1.0.1 remote-as 100</pre>	Adds an entry to the BGP neighbor table.
Step 10	neighbor <i>ip-address</i> activate Example: <pre>Router(config-router-af)# neighbor 10.1.0.1 activate</pre>	Enables the exchange of information with the peer.
Step 11	neighbor <i>ip-address</i> route-reflector-client Example: <pre>Router(config-router-af)# neighbor 10.1.0.1 route-reflector-client</pre>	Configures the local router as a BGP route reflector, and configures the specified neighbor as a route-reflector client.
Step 12	neighbor <i>ip-address</i> route-map <i>map-name</i> out Example: <pre>Router(config-router-af)# neighbor 10.1.0.1 route-map rr-out out</pre>	Applies the route map to outgoing routes from this neighbor. <ul style="list-style-type: none"> Reference the route map you created in Step 3.

	Command or Action	Purpose
Step 13	Repeat Steps 12 through 14 for the other RR clients.	You will not be applying a route map to the other RR clients.
Step 14	end Example: Router(config-router-af) # end	Exits address family configuration mode and enters privileged EXEC mode.
Step 15	show ip bgp neighbors Example: Router# show ip bgp neighbors	(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 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:

Command	Purpose
Device(config-router) # timers bgp <i>keepalive holdtime</i>	Adjusts BGP timers for all neighbors.

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in router configuration mode:

Command	Purpose
Device(config-router) # neighbor [<i>ip-address</i> <i>peer-group-name</i>] timers <i>keepalive holdtime</i>	Sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group.



Note

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 the 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:

Command	Purpose
Router(config-router)# bgp bestpath med missing-as-worst	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.

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:

Command	Purpose
Router(config-router)# bgp bestpath med confed	Configures the router to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.

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:

Command	Purpose
Router (config-router) # bgp deterministic med	Configures the router to compare the MED variable when choosing among routes advertised by different peers in the same autonomous system.



Note If the **bgp always-compare-med** router configuration command is enabled, all paths are fully comparable, including those from other autonomous systems in the confederation, even if the **bgp deterministic med** command is also enabled.

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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	<p>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	<p>bgp dampening [<i>half-life reuse suppress max-suppress-time</i>] [route-map <i>map-name</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# bgp dampening 30 1500 10000 120</pre>	<p>Enables BGP route dampening and changes the default values of route dampening factors.</p> <ul style="list-style-type: none"> • The <i>half-life</i>, <i>reuse</i>, <i>suppress</i>, and <i>max-suppress-time</i> arguments are all position dependent; if one argument is entered then all the arguments must be entered. • Use the route-map keyword and <i>map-name</i> argument to control where BGP route dampening is enabled.
Step 6	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

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:

Command	Purpose
Router# show ip bgp dampening flap-statistics	Displays BGP flap statistics for all paths.
Router# show ip bgp dampening flap-statistics regexp <i>regexp</i>	Displays BGP flap statistics for all paths that match the regular expression.
Router# show ip bgp dampening flap-statistics filter-list <i>access-list</i>	Displays BGP flap statistics for all paths that pass the filter.
Router# show ip bgp dampening flap-statistics <i>ip-address mask</i>	Displays BGP flap statistics for a single entry.
Router# show ip bgp dampening flap-statistics <i>ip-address mask</i> 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 <i>regexp</i>	Clears BGP flap statistics for all paths that match the regular expression.
Router# clear ip bgp flap-statistics filter-list <i>list</i>	Clears BGP flap statistics for all paths that pass the filter.
Router# clear ip bgp flap-statistics <i>ip-address mask</i>	Clears BGP flap statistics for a single entry.
Router# clear ip bgp <i>ip-address</i> flap-statistics	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 dampening 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 dampened-paths [<i>ip-address network-mask</i>]	Clears route dampening information and unsuppresses the suppressed routes.

Configuring BGP Route Map next-hop self

Perform this task to modify the existing route map by adding the ip next-hop self setting and overriding the bgp next-hop unchanged and bgp next-hop unchanged allpaths settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* **permit** *sequence-number*
4. **match source-protocol** *source-protocol*
5. **set ip next-hop self**
6. **exit**
7. **route-map** *map-tag* **permit** *sequence-number*
8. **match route-type internal**
9. **match route-type external**
10. **match source-protocol** *source-protocol*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpvv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **next-hop unchanged allpaths**
17. **neighbor** *ip-address* **route-map** *map-name* **out**
18. **exit**
19. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*]
20. **bgp route-map priority**
21. **redistribute** *protocol*
22. **redistribute** *protocol*
23. **exit-address-family**
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	route-map <i>map-tag</i> permit <i>sequence-number</i> Example: Device(config)# route-map static-nexthop-rewrite permit 10	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 4	match source-protocol <i>source-protocol</i> Example: Device(config-route-map)# match source-protocol static	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.
Step 5	set ip next-hop self Example: Device(config-route-map)# set ip next-hop self	Configure local routes (for BGP only) with next hop of self.
Step 6	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 7	route-map <i>map-tag</i> permit <i>sequence-number</i> Example: Device(config)# route-map static-nexthop-rewrite permit 20	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 8	match route-type internal Example: Device(config-route-map)# match route-type internal	Redistributes routes of the specified type.
Step 9	match route-type external Example: Device(config-route-map)# match route-type external	Redistributes routes of the specified type.
Step 10	match source-protocol <i>source-protocol</i> Example: Device(config-route-map)# match source-protocol connected	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.

	Command or Action	Purpose
Step 11	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor ip-address remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 172.16.232.50 remote-as 65001	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 14	address-family vpnv4 Example: Device(config-router)# address-family vpnv4	Specifies the VPNv4 address family and enters address family configuration mode.
Step 15	neighbor ip-address activate Example: Device(config-router-af)# neighbor 172.16.232.50 activate	Enables the exchange of information with a Border Gateway Protocol (BGP) neighbor.
Step 16	neighbor ip-address next-hop unchanged allpaths Example: Device(config-router-af)# neighbor 172.16.232.50 next-hop unchanged allpaths	Enables an external EBGP peer that is configured as multihop to propagate the next hop unchanged.
Step 17	neighbor ip-address route-map map-name out Example: Device(config-router-af)# neighbor 172.16.232.50 route-map static-nexthop-rewrite out	Applies a route map to an outgoing route.
Step 18	exit Example: Device(config-router-af)# exit	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 19	address-family ipv4 [unicast multicast vrf vrf-name] Example: Device(config-router)# address-family ipv4 unicast vrf inside	Specifies the IPv4 address family and enters address family configuration mode.
Step 20	bgp route-map priority Example: Device(config-router-af)# bgp route-map priority	Configures the route map priority for the local BGP routing process
Step 21	redistribute protocol Example: Device(config-router-af)# redistribute static	Redistributes routes from one routing domain into another routing domain.
Step 22	redistribute protocol Example: Device(config-router-af)# redistribute connected	Redistributes routes from one routing domain into another routing domain.
Step 23	exit-address-family Example: Device(config-router-af)# exit address-family	Exits address family configuration mode and enters router configuration mode .
Step 24	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

Configuration Examples for Internal BGP Features

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 “Example: BGP Confederation” section in this module

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.

```
router 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
 !
```

Example: 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.

```
router 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 a Next Hop for an 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
 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
```

Example: Configuring BGP Route Map next-hop self

This section contains an example of how to configure BGP Route Map next-hop self.

In this example, a route map is configured that matches the networks where you wish to override settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath`. Subsequently, `next-hop self` is configured. After this, the `bgp route map priority` is configured for the specified address family so that the previously specified route map takes priority over the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged`.

allpath. This configuration results in static routes being redistributed with a next hop of self, but connected routes and routes learned via IBGP or EBGP continue to be redistributed with an unchanged next hop.

```

route-map static-nexthop-rewrite permit 10
  match source-protocol static
  set ip next-hop self
route-map static-nexthop-rewrite permit 20
  match route-type internal
  match route-type external
  match source-protocol connected
!
router bgp 65000
  neighbor 172.16.232.50 remote-as 65001
  address-family vpnv4
    neighbor 172.16.232.50 activate
    neighbor 172.16.232.50 next-hop unchanged allpaths
    neighbor 172.16.232.50 route-map static-nexthop-rewrite out
  exit-address-family
  address-family ipv4 unicast vrf inside
    bgp route-map priority
    redistribute static
    redistribute connected
  exit-address-family
end

```

Additional References for Internal BGP Features

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module
Basic BGP configuration tasks	“Configuring a Basic BGP Network” module
iBGP multipath load sharing	“iBGP Multipath Load Sharing” module
Connecting to a service provider	“Connecting to a Service Provider Using External BGP” module
Configuring features that apply to multiple IP routing protocols	<i>IP Routing: Protocol-Independent Configuration Guide</i>

RFCs

RFC	Title
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>
RFC 1774	<i>BGP-4 Protocol Analysis</i>
RFC 1930	<i>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5396	<i>Textual Representation of Autonomous system (AS) Numbers</i>
RFC 5398	<i>Autonomous System (AS) Number Reservation for Documentation Use</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 36: Feature Information for Configuring Internal BGP Features

Feature Name	Releases	Feature Configuration Information
Configuring internal BGP features	10.3 12.0(7)T 12.0(32)S12 12.2(33)SRA 12.2(33)SXH	<p>All the features contained in this module are considered to be legacy features and will work in all trains release images.</p> <p>The following commands were introduced or modified by these features:</p> <ul style="list-style-type: none"> • bgp always-compare-med • bgp bestpath med confed • bgp bestpath med missing-as-worst • bgp client-to-client reflection • bgp cluster-id • bgp confederation identifier • bgp confederation peers • bgp dampening • bgp deterministic med • clear ip bgp dampening • clear ip bgp flap-statistics • neighbor route-reflector-client • neighbor timers • show ip bgp • show ip bgp dampening dampened-paths • show ip bgp dampening flap-statistics • timers bgp
BGP Outbound Route Map on Route Reflector to Set IP Next Hop	12.0(16)ST 12.0(22)S 12.2 12.2(14)S 15.0(1)S	<p>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.</p>



BGP VPLS Auto Discovery Support on Route Reflector

BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.

- [Finding Feature Information, page 461](#)
- [Information About BGP VPLS Auto Discovery Support on Route Reflector, page 461](#)
- [Configuration Example for BGP VPLS Auto Discovery Support on Route Reflector, page 462](#)
- [Additional References, page 462](#)
- [Feature Information for BGP VPLS Auto Discovery Support on Route Reflector, page 463](#)

Finding Feature Information

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

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

Information About BGP VPLS Auto Discovery Support on Route Reflector

BGP VPLS Autodiscovery Support on Route Reflector

In Cisco IOS Release 12.2(33)SRE, BGP VPLS Autodiscovery Support on Route Reflector was introduced. On the Cisco 7600 and Cisco 7200 series routers, 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” section. For more information about VPLS Autodiscovery, see the “VPLS Autodiscovery BGP Based” module in the *MPLS Layer 2 VPNs Configuration Guide* .

Configuration Example for BGP VPLS Auto Discovery Support on Route Reflector

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** command.

```
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
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP VPLS Auto Discovery Support on Route Reflector

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 37: Feature Information for BGP VPLS Auto Discovery Support on Route Reflector

Feature Name	Releases	Feature Information
BGP VPLS Auto Discovery Support on Route Reflector		BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.



BGP NSF Awareness

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.

- [Finding Feature Information, page 465](#)
- [Information About BGP NSF Awareness, page 466](#)
- [How to Configure BGP NSF Awareness, page 468](#)
- [Configuration Examples for BGP NSF Awareness, page 473](#)
- [Additional References, page 473](#)
- [Feature Information for BGP NSF Awareness, page 474](#)

Finding Feature Information

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

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

Information About BGP NSF Awareness

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 devices 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 module, 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 rebuilds 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 Forwarding Information Base (FIB) table and removes stale route entries. CEF then updates the line cards with the new FIB information.

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. After 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 after 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 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 an NSF-capable neighbor during an 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 to BGP peers that do not support NSF capabilities.



Note

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.

How to Configure BGP NSF Awareness

Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart

The tasks in this section show how 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 configurations of BGP NSF.

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 Bidirectional Forwarding Detection (BFD) and BGP graceful restart for NSF on a device running BGP may result in suboptimal routing.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	bgp graceful-restart [<i>restart-time seconds</i>] [<i>stalepath-time seconds</i>] Example: Device(config-router)# bgp graceful-restart	Enables the BGP graceful restart capability and BGP NSF awareness. <ul style="list-style-type: none"> • 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. • Use this command on the restarting router and all of its peers (NSF-capable and NSF-aware).
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.

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 displays 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.



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.

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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	bgp graceful-restart [<i>restart-time seconds</i>] Example: Device(config-router)# bgp graceful-restart restart-time 130	Enables the BGP graceful restart capability and BGP NSF awareness. <ul style="list-style-type: none"> • 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 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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 5	bgp graceful-restart [<i>stalepath-time seconds</i>] Example: Device(config-router)# bgp graceful-restart stalepath-time 350	Enables the BGP graceful restart capability and BGP NSF awareness. <ul style="list-style-type: none"> • 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 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 <i>Cisco IOS IP Routing: BGP Command Reference</i> .
Step 6	end Example: Device(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.

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 [*regex*] | 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
  bgp graceful-restart restart-time 130
  bgp graceful-restart stalepath-time 350
  bgp 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 [*regex*] | 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 Releases 12.2(33)SRC, 12.2(33)SB, 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.

The following partial output example using a Cisco IOS Release 12.2(33)SRC image, displays the graceful restart information for internal BGP neighbor 172.21.1.2 at Router C in the figure above. Note the “Graceful-Restart is enabled” message.

Example:

```

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-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs

```

Configuration Examples for BGP NSF Awareness

Example: Enabling BGP Global NSF Awareness Using Graceful Restart

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.

```

configure terminal
router bgp 45000
  bgp graceful-restart
  bgp graceful-restart restart-time 130
  bgp graceful-restart stalepath-time 350
end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP NSF Awareness

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 38: Feature Information for BGP NSF Awareness

Feature Name	Releases	Feature Information
BGP NSF Awareness	15.0(1)SY	<p>Nonstop Forwarding (NSF) awareness allows a device 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 device that is running BGP to forward packets along routes that are already known for a device that is performing an SSO operation. This capability allows the BGP peers of the failing device to retain the routing information that is advertised by the failing device and continue to use this information until the failed device has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.</p> <p>The following commands were introduced or modified: bgp graceful-restart, show ip bgp, show ip bgp neighbors.</p>



CHAPTER 22

IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

- [Finding Feature Information, page 477](#)
- [Information About IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family, page 477](#)
- [How to Configure IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family, page 478](#)
- [Configuration Examples for IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family, page 479](#)
- [Additional References, page 479](#)
- [Feature Information for IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family, page 480](#)

Finding Feature Information

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

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

Information About IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family

The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco nonstop forwarding (NSF) functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state.

NSF continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. Forwarding is maintained by synchronizing the FIB between the active and standby RP. On switchover, forwarding is maintained using the FIB. The RIB is not kept synchronized; therefore, the RIB is empty on switchover. The RIB is repopulated by the routing protocols and subsequently informs FIB about RIB convergence by using the NSF_RIB_CONVERGED registry call. The FIB tables are updated from the RIB, removing any stale entries. The RIB starts a failsafe timer during RP switchover, in case the routing protocols fail to notify the RIB of convergence.

The Cisco BGP address family identifier (AFI) model is designed to be modular and scalable, and to support multiple AFI and subsequent address family identifier (SAFI) configurations.

How to Configure IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Configuring the IPv6 BGP Graceful Restart Capability

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp as-number`
4. `address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6]`
5. `bgp graceful-restart [restart-time seconds | stalepath-time seconds] [all]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>router bgp as-number</code> Example: Device(config)# <code>router bgp 65000</code>	Enters router configuration mode for the specified routing process.
Step 4	<code>address-family ipv6 [vrf vrf-name] [unicast multicast vpnv6]</code>	Specifies the IPv6 address family.

	Command or Action	Purpose
	Example: Device(config-router)# address-family ipv6	
Step 5	bgp graceful-restart [restart-time <i>seconds</i> stalepath-time <i>seconds</i>] [all] Example: Device(config-router-af)# bgp graceful-restart	Enables the BGP graceful restart capability.

Configuration Examples for IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Example: Configuring the IPv6 BGP Graceful Restart Capability

In the following example, the BGP graceful restart capability is enabled:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart
```

In the following example, the restart timer is set to 130 seconds:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart restart-time 130
```

In the following example, the stalepath timer is set to 350 seconds:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart stalepath-time 350
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6: NSF and Graceful Restart for MP-BGP IPv6 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 39: Feature Information for IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family

Feature Name	Releases	Feature Information
IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family	12.2(33)SRE 15.0(1)SY 15.0(1)S 15.2(2)S	The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco NSF functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state. No commands were introduced or modified.



BGP Support for BFD

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.

- [Finding Feature Information, page 483](#)
- [Information About BGP Support for BFD, page 483](#)
- [How to Decrease BGP Convergence Time Using BFD, page 484](#)
- [Additional References, page 488](#)
- [Feature Information for BGP Support for BFD, page 489](#)

Finding Feature Information

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

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

Information About BGP Support for BFD

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.

**Caution**

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 *Cisco IOS IP Routing: BFD Configuration Guide*.

How to Decrease BGP Convergence Time Using BFD

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 software for information on configuring fast convergence.

Restrictions

- For the Cisco implementation of BFD Support for BGP in Cisco IOS Release 15.1(1)SG, only asynchronous mode is supported. In asynchronous mode, either BFD peer can initiate a BFD session.
- IPv6 encapsulation is supported.
- BFD works only for directly-connected neighbors. BFD neighbors must be no more than one IP hop away. Multihop configurations are not supported.
- Configuring both BFD and BGP Graceful Restart for NSF on a router running BGP may result in suboptimal routing.

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 place 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.

See also the “Configuring BFD for BGP IPv6 Neighbors” section in the “Configuring BGP Neighbor Session Options” module.

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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 6/0	Enters interface configuration mode.
Step 4	bfd interval <i>milliseconds</i> min_rx <i>milliseconds</i> multiplier <i>interval-multiplier</i> Example: Router(config-if)# bfd interval 50 min_rx 50 multiplier 5	Enables BFD on the interface.

	Command or Action	Purpose
Step 5	end Example: Router(config-if)# end	Exits interface configuration mode.

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.

Before You Begin

- 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 [*regex*] | dampened-routes | flap-statistics | received prefix-filter | policy [detail]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp tag1	Specifies a BGP process and enters router configuration mode.
Step 4	neighbor <i>ip-address</i> fall-over bfd Example: Router(config-router)# neighbor 172.16.10.2 fall-over bfd	Enables BFD support for fallover.
Step 5	end Example: Router(config-router)# end	Returns the router to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: Router# show bfd neighbors detail	Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip bgp neighbors [<i>ip-address</i> [received-routes routes advertised-routes paths [<i>regex</i>] dampened-routes flap-statistics received prefix-filter policy [detail]]] Example: Router# show ip bgp neighbors	Displays information about BGP and TCP connections to neighbors.

Monitoring and Troubleshooting BFD

To monitor or troubleshoot BFD, perform one or more of the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show bfd neighbors [details]**
3. **debug bfd [event | packet | ipc-error | ipc-event | oir-error | oir-event]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show bfd neighbors [details] Example: Router# show bfd neighbors details	(Optional) Displays the BFD adjacency database. <ul style="list-style-type: none"> • The details keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	debug bfd [event packet ipc-error ipc-event oir-error oir-event] Example: Router# debug bfd packet	(Optional) Displays debugging information about BFD packets.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BFD commands	Cisco IOS IP Routing: Protocol Independent Command Reference
Configuring BFD support for another routing protocol	IP Routing: BFD Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for BFD

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 40: Feature Information for BGP Support for BFD

Feature Name	Releases	Feature Information
BGP Support for BFD	12.2(33)SXH 15.0(1)SY	<p>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.</p> <p>The following commands were introduced or modified by this feature: bfd, neighbor fall-over, show bfd neighbors, and show ip bgp neighbors.</p>



BGP Support for MTR

BGP support for MTR introduces a new configuration hierarchy and command-line interface (CLI) commands to support multi-topology routing (MTR) topologies. The new configuration hierarchy, or scope, can be implemented by BGP independently of MTR. MTR allows the configuration of service differentiation through class-based forwarding. MTR supports multiple unicast topologies and a separate multicast topology. A topology is a subset of the underlying network (or base topology) characterized by an independent set of Network Layer Reachability Information (NLRI).

For more information, see the *Multitopology Routing Configuration Guide*.

- [Finding Feature Information, page 491](#)
- [Information About BGP Support for MTR, page 492](#)
- [How to Configure BGP Support for MTR, page 493](#)
- [Configuration Examples for BGP Support for MTR, page 501](#)
- [Additional References, page 504](#)
- [Feature Information for BGP Support for MTR, page 504](#)

Finding Feature Information

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

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

Information About BGP Support for MTR

BGP Network Scope

A new configuration hierarchy, named **scope**, has been introduced into the BGP protocol. To implement MTR for BGP, the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces some new configuration modes such as router scope configuration mode. Router scope configuration mode is entered by configuring the **scope** command in router configuration mode, and a collection of routing tables is created when this command is entered. BGP commands configured under the scope hierarchy are configured for a single network (globally), or on a per-VRF basis, and are referred to as **scoped commands**. The scope hierarchy can contain one or more address families.

MTR CLI Hierarchy Under BGP

The BGP CLI has been modified to provide backwards compatibility for pre-MTR BGP configuration and to provide a hierarchical implementation of MTR. Router configuration mode is backwards compatible with the pre-address family and pre-MTR configuration CLI. Global commands that affect all networks are configured in this configuration mode. For address-family and topology configuration, general session commands and peer templates can be configured to be used in the address-family or topology configuration modes.

After any global commands are configured, the scope is defined either globally or for a specific VRF. Address family configuration mode is entered by configuring the **address-family** command in router scope configuration mode or router configuration mode. Unicast is the default address family if no subaddress family (SAFI) is specified. MTR supports only the IPv4 address family with a SAFI of unicast or multicast. Entering address family configuration mode from router configuration mode configures BGP to use pre-MTR-based CLI. This configuration mode is backwards compatible with pre-existing address family configurations. Entering address family configuration mode from router scope configuration mode configures the router to use the hierarchical CLI that supports MTR. Address family configuration parameters that are not specific to a topology are entered in this address family configuration mode.

BGP topology configuration mode is entered by configuring the **topology(BGP)** command in address family configuration mode. Up to 32 topologies (including the base topology) can be configured on a router. The topology ID is configured by entering the **bgp tid** command. All address family and subaddress family configuration parameters for the topology are configured here.



Note

Configuring a scope for a BGP routing process removes CLI support for pre-MTR-based configuration.

The following shows the hierarchy levels that are used when configuring BGP for MTR implementation:

```
router bgp <
  autonomous-system-number
  >
  ! global commands

  scope {global | vrf <
    vrf-name
  >}
  ! scoped commands
```

```
    address-family {<
afi
>} [<
safi
>]
    ! address family specific commands

    topology {<
topology-name
> | base}
    ! topology specific commands
```

BGP Sessions for Class-Specific Topologies

MTR is configured under BGP on a per-session basis. The base unicast and multicast topologies are carried in the global (default) session. A separate session is created for each class-specific topology that is configured under a BGP routing process. Each session is identified by its topology ID. BGP performs a best-path calculation individually for each class-specific topology. A separate RIB and FIB are maintained for each session.

Topology Translation Using BGP

Depending on the design and policy requirements for your network, you may need to install routes from a class-specific topology on one router in a class-specific topology on a neighboring router. Topology translation functionality using BGP provides support for this operation. Topology translation is BGP neighbor-session based. The **neighbor translate-topology** command is configured using the IP address and topology ID from the neighbor.

The topology ID identifies the class-specific topology of the neighbor. The routes in the class-specific topology of the neighbor are installed in the local class-specific RIB. BGP performs a best-path calculation on all installed routes and installs these routes into the local class-specific RIB. If a duplicate route is translated, BGP will select and install only one instance of the route per standard BGP best-path calculation behavior.

Topology Import Using BGP

Topology import functionality using BGP is similar to topology translation. The difference is that routes are moved between class-specific topologies on the same router using BGP. This function is configured by entering the **import topology** command. The name of the class-specific topology or base topology is specified when entering this command. Best-path calculations are run on the imported routes before they are installed into the topology RIB. This command also includes a **route-map** keyword to allow you to filter routes that are moved between class-specific topologies.

How to Configure BGP Support for MTR

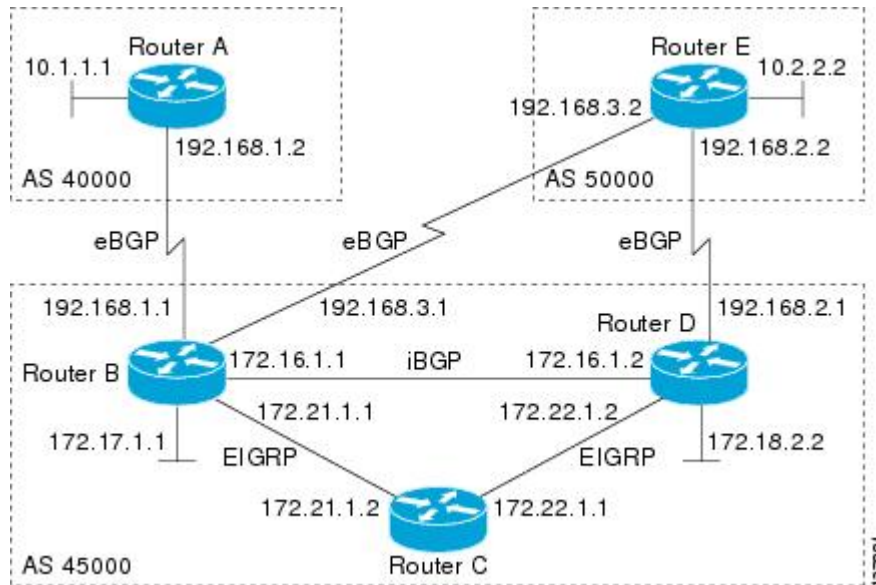
Configuring BGP Support for MTR

Before performing the following tasks, you must have configured MTR topologies. For more details, see the *"Multitopology Routing Configuration Guide."*

Activating an MTR Topology Using BGP

Perform this task to activate an MTR topology inside an address family using BGP. This task is configured on Router B in the figure below and must also be configured on Router D and Router E. In this task, a scope hierarchy is configured to apply globally and a neighbor is configured under router scope configuration mode. Under the IPv4 unicast address family, an MTR topology that applies to video traffic is activated for the specified neighbor. There is no interface configuration mode for BGP topologies.

Figure 41: BGP Network Diagram



The BGP CLI has been modified to provide backwards compatibility for pre-MTR BGP configuration and to provide a hierarchical implementation of MTR. A new configuration hierarchy, named **scope**, has been introduced into the BGP protocol. To implement MTR for BGP, the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces some new configuration modes such as router scope configuration mode. Router scope configuration mode is entered by configuring the **scope** command in router configuration mode, and a collection of routing tables is created when this command is entered. The following shows the hierarchy levels that are used when configuring BGP for MTR implementation:

```
router bgp <
  autonomous-system-number
  >
  ! global commands

  scope {global | vrf <
    vrf-name
  >}
  ! scoped commands

  address-family {<
    afi
  >} [<
    safi
  >]
  ! address family specific commands
```



```

topology {<
topology-name
> | base}
! topology specific commands

```

Before using BGP to support MTR, you should be familiar with all the concepts documented in the section, “Information About BGP Support for MTR.”

Before You Begin

- You must be running a Cisco IOS Release 12.2(33)SRB, or later release, on any routers configured for MTR.
- A global MTR topology configuration has been configured and activated.
- IP routing and CEF are enabled.



Note

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only the IPv4 address family (multicast and unicast) is supported.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **scope** {**global** | **vrf** *vrf-name*}
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **transport**{**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**}
7. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
8. **topology** {**base** | *topology-name*}
9. **bgp tid** *number*
10. **neighbor** *ip-address* **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **translate-topology** *number*
12. **end**
13. **clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**]] | **out**| **soft** [**in** [**prefix-filter**]] | **out**]]
14. **show ip bgp topology** {***** | *topology*} **summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
Step 4	<p>scope {global vrf <i>vrf-name</i>}</p> <p>Example:</p> <pre>Router(config-router)# scope global</pre>	<p>Defines the scope to the BGP routing process and enters router scope configuration mode.</p> <ul style="list-style-type: none"> • BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode. • Use the global keyword to specify that BGP uses the global routing table. • Use the vrf keyword and <i>vrf-name</i> argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 5	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-scope)# neighbor 172.16.1.2 remote-as 45000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local router.</p>
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>transport {connection-mode {active passive} path-mtu-discovery multi-session single-session}</p> <p>Example:</p> <pre>Router(config-router-scope)# neighbor 172.16.1.2 transport multi-session</pre>	<p>Enables a TCP transport session option for a BGP session.</p> <ul style="list-style-type: none"> • Use the connection-mode keyword to specify the type of connection, either active or passive. • Use the path-mtu-discovery keyword to enable TCP transport path maximum transmission unit (MTU) discovery. • Use the multi-session keyword to specify a separate TCP transport session for each address family. • Use the single-session keyword to specify that all address families use a single TCP transport session.

	Command or Action	Purpose
Step 7	<p>address-family ipv4 [mdt multicast unicast]</p> <p>Example:</p> <pre>Router(config-router-scope)# address-family ipv4</pre>	<p>Specifies the IPv4 address family and enters router scope address family configuration mode.</p> <ul style="list-style-type: none"> • Use the mdt keyword to specify IPv4 MDT address prefixes. • Use the multicast keyword to specify IPv4 multicast address prefixes. • Use the unicast keyword to specify 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. • Non-topology-specific configuration parameters are configured in this configuration mode.
Step 8	<p>topology {base topology-name}</p> <p>Example:</p> <pre>Router(config-router-scope-af)# topology VIDEO</pre>	<p>Configures the topology instance in which BGP will route class-specific or base topology traffic, and enters router scope address family topology configuration mode.</p>
Step 9	<p>bgp tid number</p> <p>Example:</p> <pre>Router(config-router-scope-af-topo)# bgp tid 100</pre>	<p>Associates a BGP routing process with the specified topology ID.</p> <ul style="list-style-type: none"> • Each topology must be configured with a unique topology ID.
Step 10	<p>neighbor ip-address activate</p> <p>Example:</p> <pre>Router(config-router-scope-af-topo)# neighbor 172.16.1.2 activate</pre>	<p>Enables the BGP neighbor to exchange prefixes for the NSAP address family with the local router.</p> <p>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.</p>
Step 11	<p>neighbor {ip-address peer-group-name} translate-topology number</p> <p>Example:</p> <pre>Router(config-router-scope-af-topo)# neighbor 172.16.1.2 translate-topology 200</pre>	<p>(Optional) Configures BGP to install routes from a topology on another router to a topology on the local router.</p> <ul style="list-style-type: none"> • The topology ID is entered for the <i>number</i> argument to identify the topology on the router.
Step 12	<p>end</p> <p>Example:</p> <pre>Router(config-router-scope-af-topo)# end</pre>	<p>(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.</p>

	Command or Action	Purpose
Step 13	<p>clear ip bgp topology <i>{* topology-name}</i> <i>{as-number dampening [network-address [network-mask]] flap-statistics [network-address [network-mask]] peer-group peer-group-name table-map update-group [number ip-address]}</i> <i>[in [prefix-filter] out soft [in [prefix-filter] out]]</i></p> <p>Example:</p> <pre>Router# clear ip bgp topology VIDEO 45000</pre>	Resets BGP neighbor sessions under a specified topology or all topologies.
Step 14	<p>show ip bgp topology <i>{* topology}</i> summary</p> <p>Example:</p> <pre>Router# show ip bgp topology VIDEO summary</pre>	<p>(Optional) Displays BGP information about a topology.</p> <ul style="list-style-type: none"> Most standard BGP keywords and arguments can be entered following the topology keyword. <p>Note Only the syntax required for this task is shown. For more details, see the <i>Cisco IOS IP Routing: BGP Command Reference</i>.</p>

Examples

The following example shows summary output for the **show ip bgp topology** command and the VIDEO topology:

```
Router# show ip bgp topology VIDEO 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
172.16.1.2    4 45000   289    289     1    0    0 04:48:44      0
192.168.3.2  4 50000     3      3     1    0    0 00:00:27      0
```

What to Do Next

Repeat this task for every topology that you want to enable, and repeat this configuration on all neighbor routers that are to use the topologies. If you want to import routes from one MTR topology to another on the same router, proceed to the next task.

Importing Routes from an MTR Topology Using BGP

Perform this task to import routes from one MTR topology to another on the same router, when multiple topologies are configured on the same router. In this task, a prefix list is defined to permit prefixes from the 10.2.2.0 network, and this prefix list is used with a route map to filter routes moved from the imported topology. A global scope is configured, address family IPv4 is entered, the VIDEO topology is specified, the VOICE topology is imported, and the routes are filtered using the route map named 10NET.

Before You Begin

- A global topology configuration has been configured and activated.
- IP routing and CEF are enabled.



Note

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops from occurring. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only the IPv4 address family (multicast and unicast) is supported.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

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-name* [**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. **scope** {**global** | **vrf** *vrf-name*}
9. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
10. **topology** {**base** | *topology-name*}
11. **import topology** {**base** | *topology-name*} [**route-map** *map-name*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] { deny <i>network/length</i> permit <i>network/length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: <pre>Router(config)# ip prefix-list TEN permit 10.2.2.0/24</pre>	Configures an IP prefix list. <ul style="list-style-type: none"> In this example, prefix list TEN permits advertising of the 10.2.2.0/24 prefix depending on a match set by the match ip address command.
Step 4	route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>] Example: <pre>Router(config)# route-map 10NET</pre>	Creates a route map and enters route map configuration mode. <ul style="list-style-type: none"> In this example, the route map named 10NET is created.
Step 5	match ip address { <i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>] } Example: <pre>Router(config-route-map)# match ip address prefix-list TEN</pre>	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. <ul style="list-style-type: none"> In this example, the route map is configured to match prefixes permitted by prefix list TEN.
Step 6	exit Example: <pre>Router(config-route-map)# exit</pre>	Exits route map configuration mode and returns to global configuration mode.
Step 7	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 50000</pre>	Enters router configuration mode to create or configure a BGP routing process.
Step 8	scope { global vrf <i>vrf-name</i> } Example: <pre>Router(config-router)# scope global</pre>	Defines the scope to the BGP routing process and enters router scope configuration mode. <ul style="list-style-type: none"> BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode.

	Command or Action	Purpose
		<ul style="list-style-type: none"> Use the global keyword to specify that BGP uses the global routing table. Use the vrf keyword and <i>vrf-name</i> argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 9	address-family ipv4 [<i>mdt</i> multicast unicast] Example: <pre>Router(config-router-scope)# address-family ipv4</pre>	Enters router scope address family configuration mode to configure an address family session under BGP. <ul style="list-style-type: none"> Non-topology-specific configuration parameters are configured in this configuration mode.
Step 10	topology { <i>base</i> <i>topology-name</i> } Example: <pre>Router(config-router-scope-af)# topology VIDEO</pre>	Configures the topology instance in which BGP will route class-specific or base topology traffic, and enters router scope address family topology configuration mode.
Step 11	import topology { <i>base</i> <i>topology-name</i> } [route-map <i>map-name</i>] Example: <pre>Router(config-router-scope-af-topo)# import topology VOICE route-map 10NET</pre>	(Optional) Configures BGP to move routes from one topology to another on the same router. <ul style="list-style-type: none"> The route-map keyword can be used to filter routes that moved between topologies.
Step 12	end Example: <pre>Router(config-router-scope-af-topo)# end</pre>	(Optional) Exits router scope address family topology configuration mode, and returns to privileged EXEC mode.

Configuration Examples for BGP Support for MTR

Example: Importing Routes from an MTR Topology Using BGP

The following example shows how to configure an access list to be used by a route map named BLUE to filter routes imported from the MTR topology named VOICE. Only routes with the prefix 192.168.1.0 are imported.

```
access-list 1 permit 192.168.1.0 0.0.0.255
route-map BLUE
  match ip address 1
  exit
router bgp 50000
  scope global
```

```

neighbor 10.1.1.2 remote-as 50000
neighbor 172.16.1.1 remote-as 60000
address-family ipv4
  topology VIDEO
  bgp tid 100
  neighbor 10.1.1.2 activate
  neighbor 172.16.1.1 activate
  import topology VOICE route-map BLUE
end
clear ip bgp topology VIDEO 50000

```

Example: Activating an MTR Topology Using BGP

This section contains the following configuration examples:

BGP Topology Translation Configuration

The following example configures BGP in the VIDEO topology and configures topology translation with the 192.168.2.2 neighbor:

```

router bgp 45000
scope global
neighbor 172.16.1.1 remote-as 50000
neighbor 192.168.2.2 remote-as 55000
neighbor 172.16.1.1 transport multi-session
neighbor 192.168.2.2 transport multi-session
address-family ipv4
  topology VIDEO
  bgp tid 100
  neighbor 172.16.1.1 activate
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 translate-topology 200
end
clear ip bgp topology VIDEO 50000

```

BGP Scope Global and VRF Configuration

The following example shows how to configure a global scope for a unicast topology and also for a multicast topology. After exiting the router scope configuration mode, a scope is configured for the VRF named DATA.

```

router bgp 45000
scope global
  bgp default ipv4-unicast
  neighbor 172.16.1.2 remote-as 45000
  neighbor 192.168.3.2 remote-as 50000
  address-family ipv4 unicast
    topology VOICE
    bgp tid 100
    neighbor 172.16.1.2 activate
  exit
  address-family ipv4 multicast
    topology base
    neighbor 192.168.3.2 activate
  exit
exit
scope vrf DATA
neighbor 192.168.1.2 remote-as 40000
address-family ipv4
neighbor 192.168.1.2 activate
end

```


BGP Topology Verification

The following example shows summary output for the **show ip bgp topology** command. Information is displayed about BGP neighbors configured to use the MTR topology named VIDEO.

```
Router# show ip bgp topology VIDEO 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
172.16.1.2    4 45000   289    289     1    0    0 04:48:44      0
192.168.3.2   4 50000     3      3     1    0    0 00:00:27      0
```

The following partial output displays BGP neighbor information under the VIDEO topology:

```
Router# show ip bgp topology VIDEO neighbors 172.16.12
```

```
BGP neighbor is 172.16.1.2, remote AS 45000, internal link
  BGP version 4, remote router ID 192.168.2.1
  BGP state = Established, up for 04:56:30
  Last read 00:00:23, last write 00:00:21, hold time is 180, keepalive interval is 60
seconds
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
  Message statistics, state Established:
    InQ depth is 0
    OutQ depth is 0

    Sent      Rcvd
  Opens:          1          1
  Notifications: 0           0
  Updates:        0           0
  Keepalives:     296        296
  Route Refresh:  0           0
  Total:          297        297
  Default minimum time between advertisement runs is 0 seconds
For address family: IPv4 Unicast topology VIDEO
  Session: 172.16.1.2 session 1
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
1 update-group member
  Topology identifier: 100
.
.
.
  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 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
  Minimum incoming TTL 0, Outgoing TTL 255
  Local host: 172.16.1.1, Local port: 11113
  Foreign host: 172.16.1.2, Foreign port: 179
.
.
.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS BGP Command Reference
MTR commands	Cisco IOS Multitopology Routing Command Reference
Configuring Multitopology Routing	Multitopology Routing Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for MTR

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 41: Feature Information for BGP Support for MTR

Feature Name	Releases	Feature Information
BGP Support for MTR	12.2(33)SRB 15.0(1)SY Cisco IOS XE Release 3.8S	<p>BGP support for MTR introduces a new configuration hierarchy and command-line interface (CLI) commands to support multi-topology routing (MTR) topologies. The new configuration hierarchy, or scope, can be implemented by BGP independently of MTR. MTR allows the configuration of service differentiation through class-based forwarding. MTR supports multiple unicast topologies and a separate multicast topology. A topology is a subset of the underlying network (or base topology) characterized by an independent set of Network Layer Reachability Information (NLRI).</p> <p>In 12.2(33)SRB, this feature was introduced on the Cisco 7600.</p> <p>The following commands were introduced or modified by this feature: address-family ipv4 (BGP), bgp tid, clear ip bgp topology, import topology, neighbor translate-topology, neighbor transport, scope, show ip bgp topology, topology (BGP).</p>



BGP Link Bandwidth

The BGP (Border Gateway Protocol) 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 507](#)
- [Prerequisites for BGP Link Bandwidth, page 508](#)
- [Restrictions for BGP Link Bandwidth, page 508](#)
- [Information About BGP Link Bandwidth, page 508](#)
- [How to Configure BGP Link Bandwidth, page 509](#)
- [Configuration Examples for BGP Link Bandwidth, page 511](#)
- [Where to Go Next, page 515](#)
- [Additional References, page 515](#)
- [Feature Information for BGP Link Bandwidth, page 516](#)

Finding Feature Information

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

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

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

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 and Verifying BGP Link Bandwidth

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4**
5. **address-family ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
6. **bgp dmzlink-bw**
7. **neighbor** *ip-address* **dmzlink-bw**
8. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
9. **end**
10. **show ip bgp** *ip-address* [**longer-prefixes** [**injected**] | **shorter-prefixes** [*mask-length*]]
11. **show ip route** *ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | [**list** *access-list-number* | *access-list-name*] | **static download**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	address-family ipv4 Example: Device(config-router-af)# address-family ipv4	Enters address family configuration mode.
Step 5	address-family ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] Example: Device(config-router)# address-family ipv4	The BGP Link Bandwidth feature is supported only under the IPv4 and VPNv4 address families.
Step 6	bgp dmzlink-bw Example: Router(config-router-af)# bgp dmzlink-bw	Configures BGP to distribute traffic proportionally to the bandwidth of the link. <ul style="list-style-type: none"> • This command must be entered on each router that contains an external interface that is to be used for multipath load balancing.
Step 7	neighbor ip-address dmzlink-bw Example: Device(config-router-af)# neighbor 172.16.1.1 dmzlink-bw	Configures BGP to include the link bandwidth attribute for routes learned from the external interface specified IP address. <ul style="list-style-type: none"> • 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 8	neighbor ip-address send-community [both extended standard] Example: Device(config-router-af)# neighbor 10.10.10.1 send-community extended	(Optional) Enables community or extended community exchange with the specified neighbor. <ul style="list-style-type: none"> • This command must be configured for iBGP peers to which the link bandwidth extended community attribute is to be propagated.
Step 9	end Example: Device(config-router-af)# end	Exits address family configuration mode, and enters privileged EXEC mode.
Step 10	show ip bgp ip-address [longer-prefixes [injected] shorter-prefixes [mask-length]]	(Optional) Displays information about the TCP and BGP connections to neighbors.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device# show ip bgp 10.0.0.0</pre>	<ul style="list-style-type: none"> The output displays the status of the link bandwidth configuration. The bandwidth of the link is shown in kilobytes.
Step 11	<p>show ip route <i>ip-address</i> [<i>mask</i>] [longer-prefixes] <i>protocol</i> [<i>process-id</i>] [list <i>access-list-number</i> <i>access-list-name</i>] static download]</p> <p>Example:</p> <pre>Device# show ip route 10.0.0.0</pre>	<p>(Optional) Displays the current state of the routing table.</p> <ul style="list-style-type: none"> The output displays traffic share values, including the weights of the links that are used to direct traffic proportionally to the bandwidth of each link.

Configuration Examples for BGP Link Bandwidth

Example: BGP Link Bandwidth Configuration

In the following examples, the BGP Link Bandwidth feature is configured so BGP will distribute traffic proportionally to the bandwidth of each external link. The figure below shows two external autonomous 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.



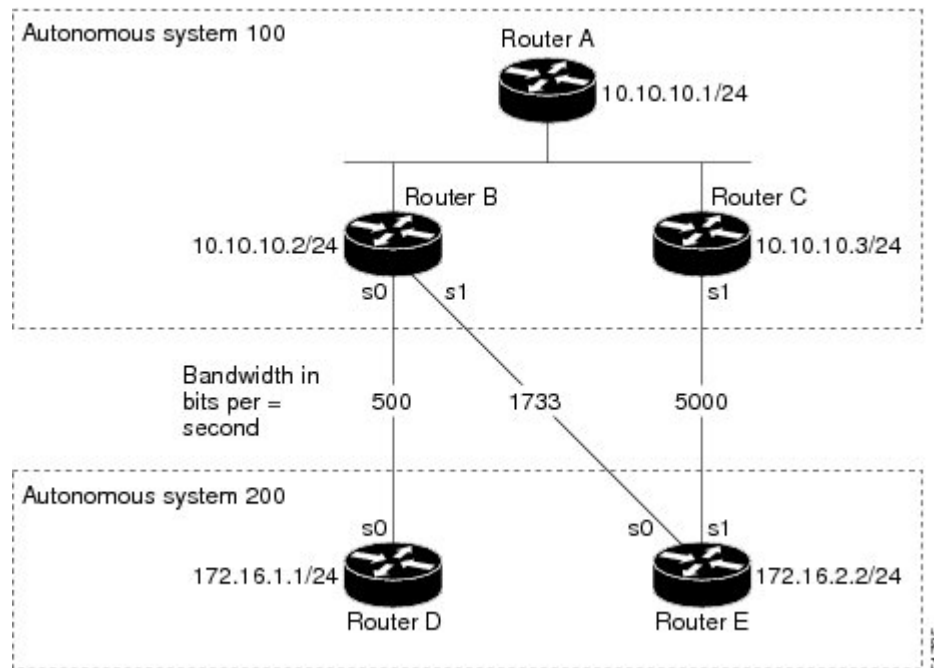
Note

The BGP Link Bandwidth feature functions for simple topologies that have a single path toward the exit points.

**Caution**

The BGP Link Bandwidth feature might not function properly if load balancing is required toward the exit points.

Figure 42: BGP Link Bandwidth Configuration



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:

```
RouterA(config)# router bgp 100
RouterA(config-router)# neighbor 10.10.10.2 remote-as 100
RouterA(config-router)# neighbor 10.10.10.2 update-source Loopback 0
RouterA(config-router)# neighbor 10.10.10.3 remote-as 100
RouterA(config-router)# neighbor 10.10.10.3 update-source Loopback 0
RouterA(config-router)# address-family ipv4
RouterA(config-router-af)# bgp dmzlink-bw
RouterA(config-router-af)# neighbor 10.10.10.2 activate
RouterA(config-router-af)# neighbor 10.10.10.2 send-community both
RouterA(config-router-af)# neighbor 10.10.10.3 activate
RouterA(config-router-af)# neighbor 10.10.10.3 send-community both
RouterA(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:

```
RouterB(config)# router bgp 100
RouterB(config-router)# neighbor 10.10.10.1 remote-as 100
RouterB(config-router)# neighbor 10.10.10.1 update-source Loopback 0
RouterB(config-router)# neighbor 10.10.10.3 remote-as 100
RouterB(config-router)# neighbor 10.10.10.3 update-source Loopback 0
RouterB(config-router)# neighbor 172.16.1.1 remote-as 200
RouterB(config-router)# neighbor 172.16.1.1 ebgp-multihop 1
RouterB(config-router)# neighbor 172.16.2.2 remote-as 200
RouterB(config-router)# neighbor 172.16.2.2 ebgp-multihop 1
RouterB(config-router)# address-family ipv4
RouterB(config-router-af)# bgp dmzlink-bw
RouterB(config-router-af)# neighbor 10.10.10.1 activate
RouterB(config-router-af)# neighbor 10.10.10.1 next-hop-self
RouterB(config-router-af)# neighbor 10.10.10.1 send-community both
RouterB(config-router-af)# neighbor 10.10.10.3 activate
RouterB(config-router-af)# neighbor 10.10.10.3 next-hop-self
RouterB(config-router-af)# neighbor 10.10.10.3 send-community both
RouterB(config-router-af)# neighbor 172.16.1.1
RouterB(config-router-af)# activate
RouterB(config-router-af)# neighbor 172.16.1.1 dmzlink-bw
RouterB(config-router-af)# neighbor 172.16.2.2 activate
RouterB(config-router-af)# neighbor 172.16.2.2 dmzlink-bw
RouterB(config-router-af)# maximum-paths ibgp 6
RouterB(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:

```
RouterC(config)# router bgp 100
RouterC(config-router)# neighbor 10.10.10.1 remote-as 100
RouterC(config-router)# neighbor 10.10.10.1 update-source Loopback 0
RouterC(config-router)# neighbor 10.10.10.2 remote-as 100
RouterC(config-router)# neighbor 10.10.10.2 update-source Loopback 0
RouterC(config-router)# neighbor 172.16.3.30 remote-as 200
RouterC(config-router)# neighbor 172.16.3.30 ebgp-multihop 1
RouterC(config-router)# address-family ipv4
RouterC(config-router-af)# bgp dmzlink-bw
RouterC(config-router-af)# neighbor 10.10.10.1 activate
RouterC(config-router-af)# neighbor 10.10.10.1 send-community both
RouterC(config-router-af)# neighbor 10.10.10.1 next-hop-self
RouterC(config-router-af)# neighbor 10.10.10.2 activate
RouterC(config-router-af)# neighbor 10.10.10.2 send-community both
```

```

RouterC(config-router-af)# neighbor 10.10.10.2 next-hop-self
RouterC(config-router-af)# neighbor 172.16.3.3 activate
RouterC(config-router-af)# neighbor 172.16.3.3 dmzlink-bw

RouterC(config-router-af)# maximum-paths ibgp 6
RouterC(config-router-af)# maximum-paths 6

```

Example: Verifying BGP Link Bandwidth

The examples in this section show the verification of this feature on Router A, Router B, and Router C.

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.

```

RouterB# 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. Exit links are located on Router B and Router C. The output shows that a route for each exit link to autonomous system 200 has been installed as a best path in the BGP routing table.

```

RouterA# 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:

```
RouterA# 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
```

Where to Go Next

For information about the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN feature, see the “BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN” module in the *IP Routing: BGP Configuration Guide*.

For information about the iBGP Multipath Load Sharing feature, see the “iBGP Multipath Load Sharing” module in the *IP Routing: BGP Configuration Guide*.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
CEF configuration tasks	<i>IP Switching Cisco Express Forwarding Configuration Guide</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFC	Title
draft-ramachandra-bgp-ext-communities-09.txt	<i>BGP Extended Communities Attribute</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 42: Feature Information for BGP Link Bandwidth

Feature Name	Releases	Feature Information
BGP Link Bandwidth	12.2(2)T 12.2(14)S	<p>This feature advertises the bandwidth of an autonomous system exit link as an extended community. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled.</p> <p>The following commands were introduced or modified: router bgp, address-family ipv4, address-family ipv4, bgp dmzlink-bw, neighbor, show ip bgp, show ip route.</p>



iBGP Multipath Load Sharing

This feature module describes the iBGP Multipath Load Sharing feature.

- [Finding Feature Information, page 519](#)
- [Restrictions for iBGP Multipath Load Sharing, page 519](#)
- [Information about iBGP Multipath Load Sharing, page 520](#)
- [How to Configure iBGP Multipath Load Sharing, page 522](#)
- [Configuration Examples for iBGP Multipath Load Sharing, page 525](#)
- [Additional References, page 526](#)
- [Feature Information for iBGP Multipath Load Sharing, page 528](#)

Finding Feature Information

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

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

Restrictions for iBGP Multipath Load Sharing

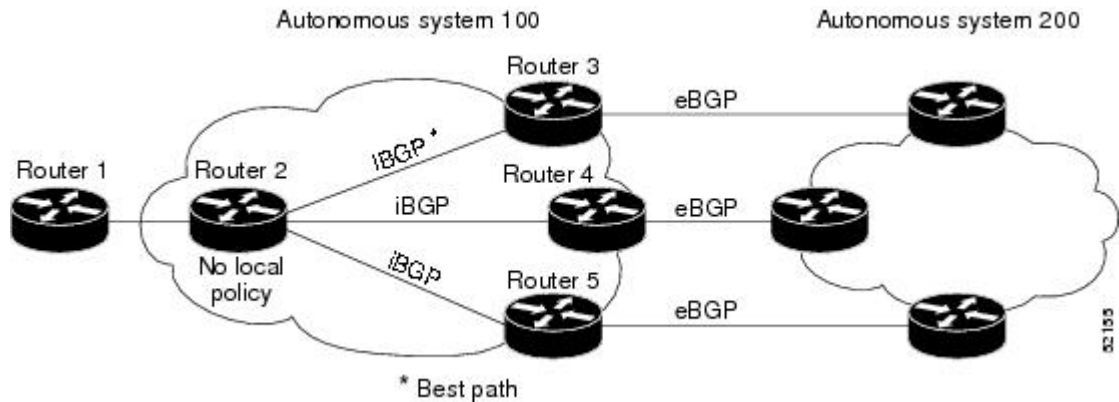
- **Route Reflector Limitation**—With multiple iBGP paths installed in a routing table, a route reflector will advertise only one of the paths (one next hop).
- **Memory Consumption Restriction**—Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses approximately 350 bytes of 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.

Information about iBGP Multipath Load Sharing

iBGP Multipath Load Sharing Overview

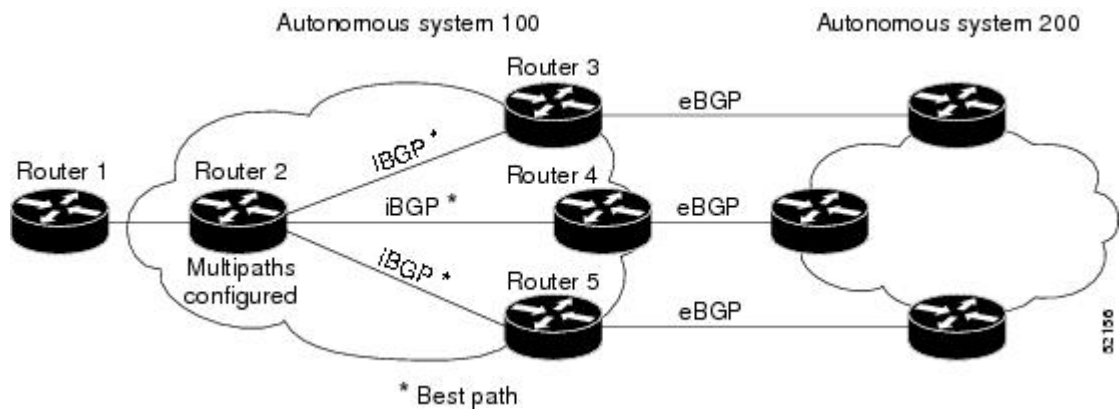
When a Border Gateway Protocol (BGP) speaking router with no local policy configured receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router will choose one iBGP path as the best path. The best path is then installed in the IP 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 43: Non-MPLS Topology with One Best Path



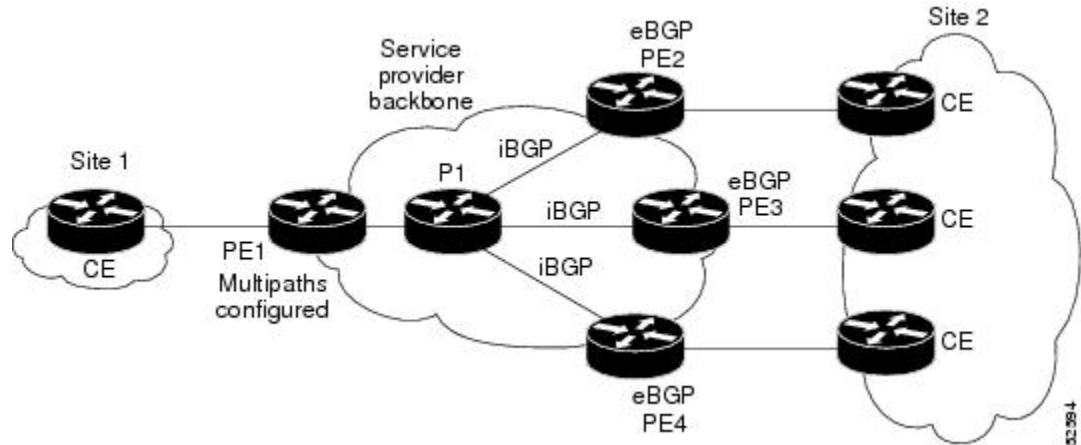
The iBGP Multipath Load Sharing feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router. For example, on router 2 in the figure below, the paths to routers 3, 4, and 5 are configured as multipaths and can be used to reach autonomous system 200, thereby equally sharing the load to autonomous system 200.

Figure 44: Non-MPLS Topology with Three Multipaths



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 45: 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.

The iBGP Multipath Load Sharing feature is similar to BGP multipath support for external BGP (eBGP) paths; however, the iBGP Multipath Load Sharing feature is applied to internal rather than eBGP paths.

Benefits of iBGP Multipath Load Sharing

Configuring multiple iBGP best paths enables a router to evenly share the traffic destined for a particular site.

How to Configure iBGP Multipath Load Sharing

Configuring iBGP Multipath Load Sharing

To configure the iBGP Multipath Load Sharing feature, use the following command in router configuration mode:

Command	Purpose
Device (config-router) # maximum-paths ibgp <i>maximum-number</i>	Controls the maximum number of parallel iBGP routes that can be installed in a routing 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 vpnv4 all ip-prefix** EXEC command to display attributes for a network in an MPLS VPN:
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."
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 vpnv4 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 vpnv4 all ip-prefix** EXEC command to display attributes for a network in an MPLS VPN:

Example:

```
Device# 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
Flag:0x820
  Advertised to non peer-group peers:
```

```

10.1.12.12
22
  10.2.3.8 (metric 11) from 10.1.3.4 (100.0.0.5)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
    Originator:100.0.0.5, Cluster list:100.0.0.4
22
  10.2.1.9 (metric 11) from 10.1.1.2 (100.0.0.9)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath
    Originator:100.0.0.9, Cluster list:100.0.0.2
22
  10.2.5.10 (metric 11) from 10.1.5.6 (100.0.0.10)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath
    Originator:100.0.0.10, Cluster list:100.0.0.6
22
  10.2.4.10 (metric 11) from 10.1.4.5 (100.0.0.10)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath
    Originator:100.0.0.10, Cluster list:100.0.0.5
22
  10.2.6.10 (metric 11) from 10.1.6.7 (100.0.0.10)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath
    Originator:100.0.0.10, Cluster list:100.0.0.7

```

```
Device# show ip bgp vpnv4 all 10.22.22.0
```

```

BGP routing table entry for 100:1:10.22.22.0/24, version 50
Paths:(6 available, best #1)
Multipath:iBGP
  Advertised to non peer-group peers:
  200.1.12.12
  22
    10.22.7.8 (metric 11) from 10.11.3.4 (100.0.0.8)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
      Extended Community:RT:100:1
      Originator:100.0.0.8, Cluster list:100.1.1.44
  22
    10.22.1.9 (metric 11) from 10.11.1.2 (100.0.0.9)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.9, Cluster list:100.1.1.22
  22
    10.22.6.10 (metric 11) from 10.11.6.7 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.7
  22
    10.22.4.10 (metric 11) from 10.11.4.5 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.5
  22
    10.22.5.10 (metric 11) from 10.11.5.6 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.6

```

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:

```

Device# 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
Last update from 10.2.6.10 00:00:03 ago
Routing Descriptor Blocks:
* 10.2.3.8, from 10.1.3.4, 00:00:03 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
  10.2.1.9, from 10.1.1.2, 00:00:03 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.2.5.10, from 10.1.5.6, 00:00:03 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.2.4.10, from 10.1.4.5, 00:00:03 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.2.6.10, from 10.1.6.7, 00:00:03 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1

```

```
Device# show ip route vrf PATH 10.22.22.0
```

```

Routing entry for 10.22.22.0/24
Known via "bgp 1", distance 200, metric 0
Tag 22, type internal
Last update from 10.22.5.10 00:01:07 ago
Routing Descriptor Blocks:
* 10.22.7.8 (Default-IP-Routing-Table), from 10.11.3.4, 00:01:07 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
  10.22.1.9 (Default-IP-Routing-Table), from 10.11.1.2, 00:01:07 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.22.6.10 (Default-IP-Routing-Table), from 10.11.6.7, 00:01:07 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.22.4.10 (Default-IP-Routing-Table), from 10.11.4.5, 00:01:07 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.22.5.10 (Default-IP-Routing-Table), from 10.11.5.6, 00:01:07 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1

```

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 vpv4 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:

Command	Purpose
Device# <code>show ip bgp ip-prefix</code>	Displays attributes and multipaths for a network in a non-MPLS topology.
Device# <code>show ip bgp vpnv4 all ip-prefix</code>	Displays attributes and multipaths for a network in an MPLS VPN.
Device# <code>show ip route ip-prefix</code>	Displays routing information for a network in a non-MPLS topology.
Device# <code>show ip route vrf vrf-name ip-prefix</code>	Displays routing information for a network in an MPLS VPN.

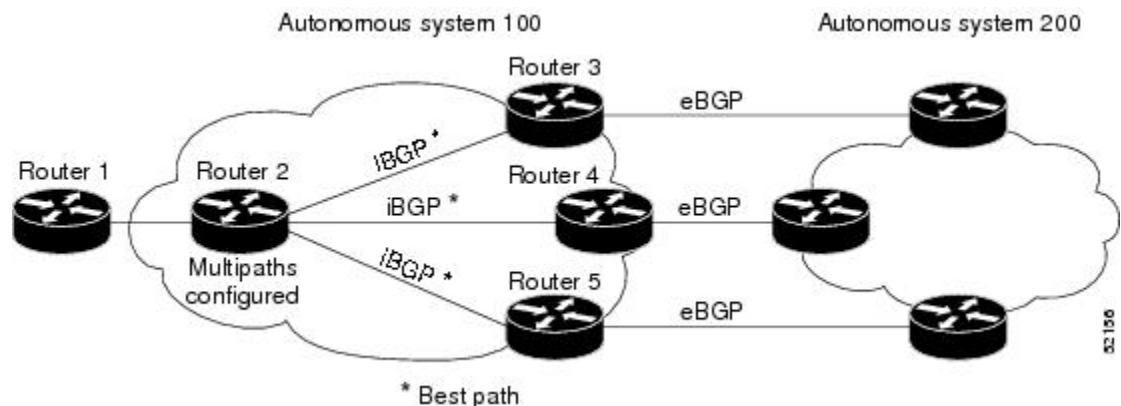
Configuration Examples for iBGP Multipath Load Sharing

Example: iBGP Multipath Load Sharing in a Non-MPLS Topology

Both examples assume that the appropriate attributes for each path are equal and that the next hop router for each multipath is different.

The following example shows how to set up the iBGP Multipath Load Sharing feature in a non-MPLS topology (see the figure below).

Figure 46: Non-MPLS Topology Example



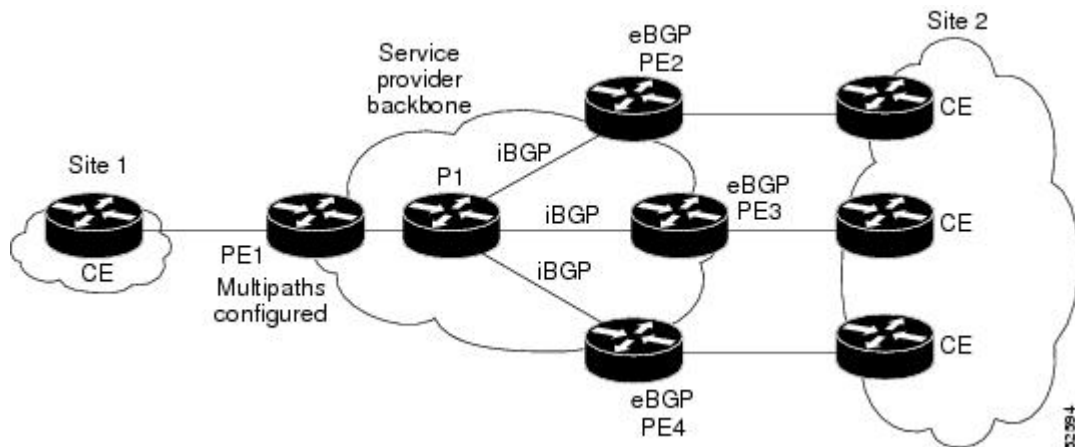
Router 2 Configuration

```
router 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 47: MPLS VPN Topology Example

**Router PE1 Configuration**

```
router bgp 100
address-family ipv4 unicast vrf site2
maximum-paths ibgp 3
```

Additional References**Related Documents**

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
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” module in the <i>IP Routing: BGP Configuration Guide</i>
Advertising the bandwidth of an autonomous system exit link as an extended community	“BGP Link Bandwidth” module in the <i>IP Routing: BGP Configuration Guide</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
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 XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/cisco/tools/cfn.html](#). An account on Cisco.com is not required.

Table 43: Feature Information for iBGP Multipath Load Sharing

Feature Name	Releases	Feature Information
iBGP Multipath Load Sharing	12.2(14)S 12.2(2)T	<p>The iBGP Multipath Load Sharing feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination.</p> <p>The following command was introduced: maximum-paths ibgp</p> <p>The following commands were modified: show ip bgp, show ip bgp vpnv4, show ip route, and show ip route vrf.</p>



CHAPTER 27

BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for eBGP and iBGP 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 multi-homed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

- [Finding Feature Information, page 529](#)
- [Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 530](#)
- [Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 530](#)
- [Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 530](#)
- [How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 532](#)
- [Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature, page 535](#)
- [Where to Go Next, page 536](#)
- [Additional References, page 536](#)
- [Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 537](#)

Finding Feature Information

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

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

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

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.



Note

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.



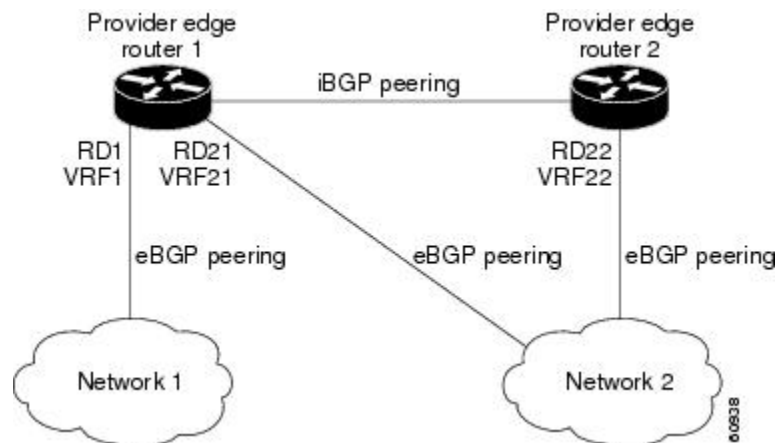
Note

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 48: A Service Provider BGP MPLS Network

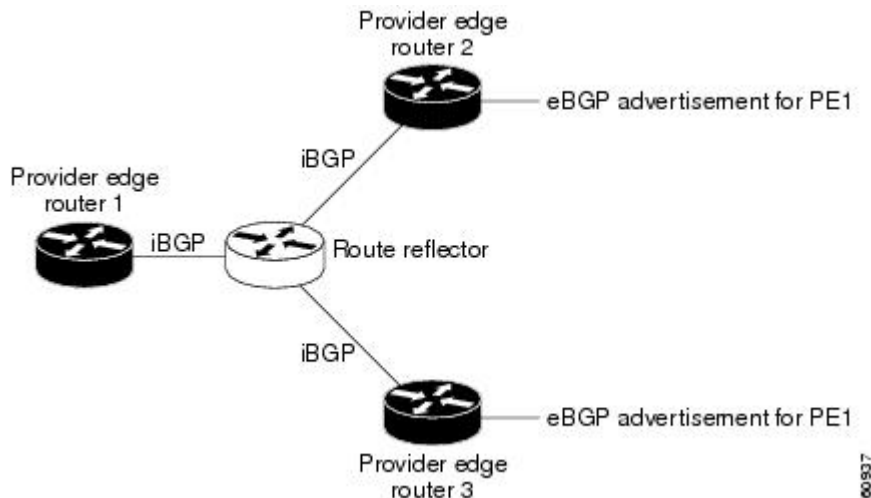


PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF of Network 1. The multipaths will be used by CEF to perform load balancing. IP traffic that is sent 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 to PE router 1.

Figure 49: 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 and 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*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	address-family ipv4 vrf <i>vrf-name</i> Example: Device(config-router)# address-family ipv4 vrf RED	Places the router in address family configuration mode. • Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 5	maximum-paths eibgp <i>number</i> Example: Device(config-router-af)# maximum-paths eibgp 6	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. 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.
Step 6	end Example: Device(config-router-af)# end	Exits address family configuration mode, and enters privileged EXEC mode.

Verifying Multipath Load Sharing for Both eBGP and iBGP

SUMMARY STEPS

1. `enable`
2. `show ip bgp neighbors [neighbor-address [advertised-routes | dampened-routes | flap-statistics | paths [regex] | 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

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>show ip bgp neighbors [neighbor-address [advertised-routes dampened-routes flap-statistics paths [regex] received prefix-filter received-routes routes]]</code></p> <p>Example:</p> <pre>Device# show ip bgp neighbors</pre>	<p>Displays information about the TCP and BGP connections to neighbors.</p>
Step 3	<p><code>show ip bgp vpnv4 {all rd route-distinguisher vrf vrf-name}</code></p> <p>Example:</p> <pre>Device# show ip bgp vpnv4 vrf RED</pre>	<p>Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP.</p>
Step 4	<p><code>show ip route vrf vrf-name</code></p> <p>Example:</p> <pre>Device# show ip route vrf RED</pre>	<p>Displays the IP routing table associated with a VRF instance. The <code>show ip route vrf</code> command is used to verify that the VRF is in the routing table.</p>

Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature

Example: Configuring eBGP and iBGP Multipath Load Sharing

This following configuration example configures a router in address-family mode to select six BGP routes (eBGP or iBGP) as multipaths:

```
Device(config)# router bgp 40000
Device(config-router)# address-family ipv4 vrf RED
Device(config-router-af)# maximum-paths eibgp 6
Device(config-router-af)# end
```

Example: Verifying eBGP and iBGP Multipath Load Sharing

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:

```
Device# 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)
      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:10.0.0.3
    22
    10.1.1.12 from 10.1.1.12 (10.22.22.12)
      Origin IGP, metric 0, localpref 100, valid, external, multipath, best
      Extended Community:RT:100:1
```

In the following example, the **show ip route vrf** command is entered to display multipath routes in the VRF table:

```
Device# 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 10.0.0.2, 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” module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Comprehensive BGP link bandwidth configuration examples and tasks	“BGP Link Bandwidth” module in the <i>IP Routing: BGP Configuration Guide</i>
CEF configuration tasks	“CEF Overview” module in the <i>IP Switching Cisco Express Forwarding Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 1771	<i>A Border Gateway Protocol 4 (BGP4)</i>
RFC 2547	<i>BGP/MPLS VPNs</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 44: Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Feature Name	Releases	Feature Configuration Information
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	12.0(24)S 12.2(14)S 12.2(18)SXE 12.2(4)T 15.0(1)S Cisco IOS XE Release 3.1.0SG	<p>The BGP Multipath Load Sharing for eBGP and iBGP feature allows you to configure multipath load balancing with both eBGP and iBGP paths in BGP networks that are configured to use MPLS VPNs. This feature provides improved load balancing deployment and service offering capabilities and is useful for multi-homed autonomous systems and PE routers that import both eBGP and iBGP paths from multihomed and stub networks.</p> <p>The following command was introduced or modified by this feature: maximum-paths eibgp.</p>



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 for multipath loadsharing.

- [Finding Feature Information](#), page 539
- [Overview of Loadsharing IP Packets over More Than Six Parallel Paths](#), page 539
- [Additional References](#), page 540
- [Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths](#), page 541

Finding Feature Information

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

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

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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
eBGP multipath load sharing	“BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN” module
iBGP multipath load sharing	“iBGP Multipath Load Sharing” module

MIBs

MIB	MIBs Link
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 XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 45: Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths

Feature Name	Releases	Feature Information
Loadsharing IP Packets Over More Than Six Parallel Paths	12.3(2)T 12.2(25)S Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.1.0SG	<p>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 for multipath loadsharing.</p> <p>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</p> <p>The following commands were modified:</p> <ul style="list-style-type: none"> • maximum-paths • maximum-paths eibgp • maximum-paths ibgp • show ip route summary



BGP Policy Accounting

Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and 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 543](#)
- [Prerequisites, page 543](#)
- [Information About BGP Policy Accounting, page 544](#)
- [How to Configure BGP Policy Accounting, page 545](#)
- [Configuration Examples for BGP Policy Accounting, page 549](#)
- [Additional References, page 550](#)
- [Feature Information for BGP Policy Accounting, page 551](#)

Finding Feature Information

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

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

Prerequisites

Before using the BGP Policy Accounting feature, you must enable BGP and CEF or dCEF on the router.

Information About BGP Policy Accounting

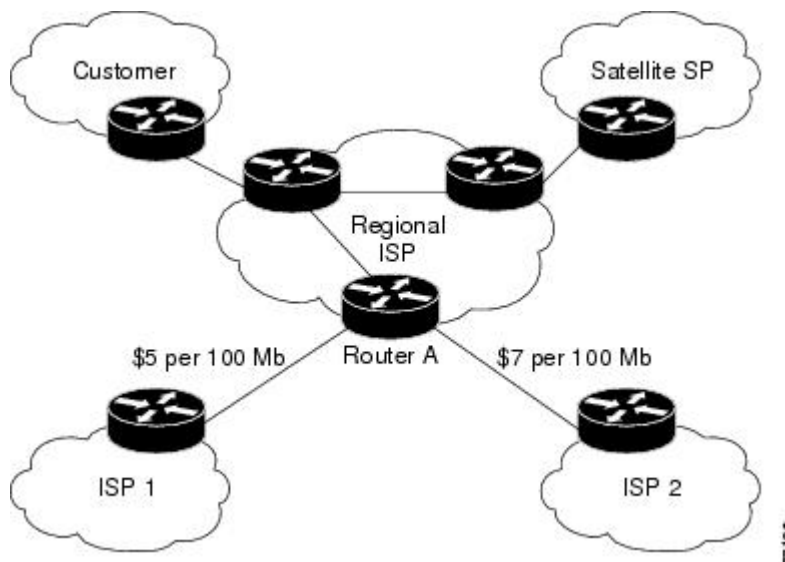
BGP Policy Accounting Overview

Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and 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 interface. A Cisco IOS policy-based classifier maps the traffic into one of eight possible buckets, representing different traffic classes.

Using BGP policy accounting, you can account for traffic according to the route it traverses. Service providers (SPs) can identify and account for all traffic by customer and bill accordingly. In the figure below, BGP policy accounting 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 50: 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 Policy Accounting

Account 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. Service providers can account for traffic and apply billing, according to the route 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 Policy Accounting

Specifying the Match Criteria for BGP Policy Accounting

The first task in configuring BGP policy accounting 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.

To specify the BGP attribute to use for BGP policy accounting and create the match criteria in a route map, use the following commands in global configuration mode:

SUMMARY STEPS

1. Device(config)# **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*
2. Device(config)# **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
3. Device(config-route-map)# **match community-list** *community-list-number* [**exact**]
4. Device(config-route-map)# **set traffic-index** *bucket-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# ip community-list <i>community-list-number</i> { permit deny } <i>community-number</i>	Creates a community list for BGP and controls access to it. This step must be repeated for each community to be specified.
Step 2	Device(config)# route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]	Enters route-map configuration mode and defines the conditions for policy routing. The <i>map-name</i> argument identifies a route map. The optional permit and deny keywords work with the match and set criteria to control how the packets are accounted for.

	Command or Action	Purpose
		The optional <i>sequence-number</i> argument indicates the position a new route map is to have in the list of route maps already configured with the same name.
Step 3	Device(config-route-map)# match community-list <i>community-list-number</i> [exact]	Matches a BGP community.
Step 4	Device(config-route-map)# set traffic-index <i>bucket-number</i>	Indicates where to output packets that pass a match clause of a route map for BGP policy accounting.

Classifying the IP Traffic and Enabling BGP Policy Accounting

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 it adds to the routing table based on the match criteria. When the **bgp-policy accounting** command is configured on an interface, BGP policy accounting is enabled.

To classify the IP traffic and enable BGP policy accounting, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Device(config)# **router bgp** *as-number*
2. Device(config-router)# **table-map** *route-map-name*
3. Device(config-router)# **network** *network-number* [**mask** *network-mask*]
4. Device(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
5. Device(config-router)# **exit**
6. Device(config)# **interface** *interface-type* *interface-number*
7. Device(config-if)# **no ip directed-broadcast**
8. Device(config-if)# **ip address** *ip-address* *mask*
9. Device(config-if)# **bgp-policy accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# router bgp <i>as-number</i>	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 2	Device(config-router)# table-map <i>route-map-name</i>	Classifies BGP prefixes entered in the routing table.

	Command or Action	Purpose
Step 3	Device(config-router)# network <i>network-number</i> [mask <i>network-mask</i>]	Specifies a network to be advertised by the BGP routing process.
Step 4	Device(config-router)# neighbor <i>ip-address</i> remote-as <i>as-number</i>	Specifies a BGP peer by adding an entry to the BGP routing table.
Step 5	Device(config-router)# exit	Exits to global configuration mode.
Step 6	Device(config)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interface type and number and enters interface configuration mode.
Step 7	Device(config-if)# no ip directed-broadcast	Configures the interface to drop directed broadcasts destined for the subnet to which that interface is attached, rather than being broadcast. This is a security issue.
Step 8	Device(config-if)# ip address <i>ip-address mask</i>	Configures the interface with an IP address.
Step 9	Device(config-if)# bgp-policy accounting	Enables BGP policy accounting for the interface.

Verifying BGP Policy Accounting

To verify that BGP policy accounting is operating, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip cef EXEC** command with the **detail** keyword to learn which accounting bucket is assigned to a specified prefix.
2. Enter the **show ip bgp EXEC** command for the same prefix used in Step 1--192.168.5.0-- to learn which community is assigned to this prefix.
3. Enter the **show cef interface policy-statistics EXEC** command to display the per-interface traffic statistics.

DETAILED STEPS

- Step 1** Enter the **show ip cef EXEC** 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 the accounting bucket number 4 (traffic_index 4) is assigned to this prefix.

Example:

```
Device# 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
  via 10.14.1.1, 0 dependencies, recursive
```

```
next hop 10.14.1.1, POS7/2 via 10.14.1.0/30
valid cached adjacency
```

Step 2 Enter the **show ip bgp EXEC** 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:

```
Device# 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 Enter the **show cef interface policy-statistics EXEC** 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:

```
Device# show cef interface policy-statistics

POS7/0 is up (if_number 8)
Bucket   Packets   Bytes
1         0         0
2         0         0
3         50        5000
4         100       10000
5         100       10000
6         10        1000
7         0         0
8         0         0
```

Monitoring and Maintaining BGP Policy Accounting

Command	Purpose
Device# show cef interface [<i>type number</i>] policy-statistics	(Optional) Displays detailed CEF policy statistical information for all interfaces.
Device# show ip bgp [<i>network</i>] [<i>network mask</i>] [longer-prefixes]	(Optional) Displays entries in the BGP routing table.
Device# show ip cef [<i>network</i> [<i>mask</i>]] [detail]	(Optional) Displays entries in the Forwarding Information Base (FIB) or FIB summary information.

Configuration Examples for BGP Policy Accounting

Example: Specifying the Match Criteria for BGP Policy Accounting

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 30
set traffic-index 2
!
route-map set_bucket permit 20
match community 40
set traffic-index 3
!
route-map set_bucket permit 30
match community 50
set traffic-index 4
!
route-map set_bucket permit 40
match community 60
set traffic-index 5
```

Example: Classifying the IP Traffic and Enabling BGP Policy Accounting

In the following example, BGP policy accounting is enabled on POS interface 7/0 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 POS7/0
 ip address 10.15.1.2 255.255.255.0
 no ip directed-broadcast
 bgp-policy accounting
 no keepalive
 crc 32
 clock source internal
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) commands	Cisco IOS IP Switching Command Reference
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) configuration information	“CEF Overview” module of the <i>Cisco IOS Switching Services Configuration Guide</i>

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> • CISCO-BGP-POLICY-ACCOUNTING-MIB <p>Note CISCO-BGP-POLICY-ACCOUNTING-MIB is only available in the Cisco IOS Release 12.0(9)S, 12.0(17)ST, and later releases. This MIB is not available on any mainline and T-train release.</p>	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Policy 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 46: Feature Information for BGP Policy Accounting

Feature Name	Releases	Feature Information
BGP Policy Accounting	12.0(9)S 12.0(17)ST 12.2(13)T 15.0(1)S 12.2(50)SY Cisco IOS XE Release 3.8S	<p>Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.</p> <p>The following commands were introduced or modified:</p> <ul style="list-style-type: none"> • bgp-policy • set traffic-index • show cef interface policy-statistics • show ip bgp • show ip cef



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 553](#)
- [Prerequisites for BGP PA Output Interface Accounting, page 554](#)
- [Information About BGP PA Output Interface Accounting, page 554](#)
- [How to Configure BGP PA Output Interface Accounting, page 555](#)
- [Configuration Examples for BGP PA Output Interface Accounting, page 562](#)
- [Additional References, page 562](#)
- [Feature Information for BGP Policy Accounting Output Interface Accounting, page 564](#)

Finding Feature Information

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

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

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

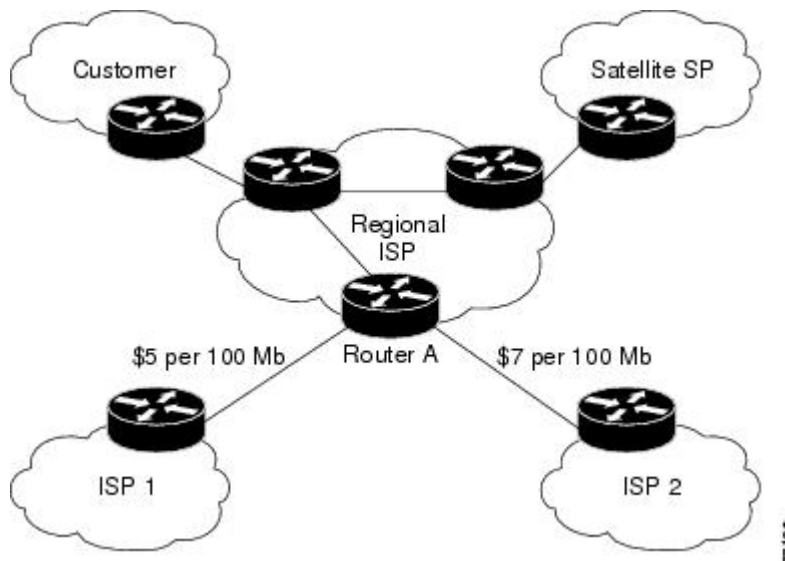
BGP PA Output Interface Accounting

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 51: 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

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-number* **[exact]**
6. **set traffic-index** *bucket-number*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>ip community-list <i>{standard-list-number expanded-list-number [regular-expression]}</i> {standard expanded} <i>community-list-name</i> {permit deny} <i>{community-number regular-expression}</i></p> <p>Example:</p> <pre>Device(config)# ip community-list 30 permit 100:190</pre>	<p>Creates a community list for BGP and controls access to it.</p> <ul style="list-style-type: none"> • Repeat this step for each community to be specified.
Step 4	<p>route-map <i>map-name</i> [permit deny] <i>[sequence-number]</i></p> <p>Example:</p> <pre>Device(config)# route-map set_bucket permit 10</pre>	<p>Enters route-map configuration mode and defines the conditions for policy routing.</p> <ul style="list-style-type: none"> • The <i>map-name</i> argument identifies a route map. • The optional permit and deny keywords work with the match and set criteria to control how the packets are accounted for. • The optional <i>sequence-number</i> argument indicates the position that a new route map is to have in the list of route maps already configured with the same name.
Step 5	<p>match community-list <i>community-list-number</i> [exact]</p> <p>Example:</p> <pre>Router(config-route-map)# match community-list 30</pre>	Matches a BGP community.
Step 6	<p>set traffic-index <i>bucket-number</i></p> <p>Example:</p> <pre>Device(config-route-map)# set traffic-index 2</pre>	Indicates where to output packets that pass a match clause of a route map for BGP policy accounting.
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route-map configuration mode and returns to global configuration mode.

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode for the specified routing process. <ul style="list-style-type: none"> • The <i>as-number</i> argument identifies a BGP autonomous system number.

	Command or Action	Purpose
Step 4	table-map <i>route-map-name</i> Example: Device(config-router)# table-map set_bucket	Classifies BGP prefixes entered in the routing table.
Step 5	network <i>network-number</i> [mask <i>network-mask</i>] Example: Device(config-router)# network 10.15.1.0 mask 255.255.255.0	Specifies a network to be advertised by the BGP routing process.
Step 6	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: Device(config-router)# neighbor 10.14.1.1 remote-as 65100	Specifies a BGP peer by adding an entry to the BGP routing table.
Step 7	exit Example: Device(config-router)# exit	Exits router configuration mode and returns to global configuration mode.
Step 8	interface <i>type number</i> Example: Device(config)# interface POS 7/0	Specifies the interface type and number and enters interface configuration mode. <ul style="list-style-type: none"> • The <i>type</i> argument identifies the type of interface. • The <i>number</i> argument identifies the slot and port numbers of the interface. The space between the interface type and number is optional.
Step 9	ip address <i>ip-address mask</i> Example: Device(config-if)# ip-address 10.15.1.2 255.255.255.0	Configures the interface with an IP address.
Step 10	bgp-policy accounting [input output] [source] Example: Device(config-if)# bgp-policy accounting input source	Enables BGP policy accounting for the interface. <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
Step 11	exit Example: Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.

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:

```
Device# 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
  via 10.14.1.1, 0 dependencies, recursive
  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:

```
Device# 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*]

Displays 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:

```
Device# 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

```

Index	Packets	Bytes
1	9999	999900
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	1234	123400
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0
41	0	0
42	0	0
43	0	0
44	0	0
45	1000	100000
46	0	0

47	0	0
48	0	0
49	0	0
50	0	0
51	0	0
52	0	0
53	0	0
54	5123	1198782
55	0	0
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0	0
62	0	0
63	0	0
64	0	0

Step 4 **show cef interface** [*type number*] [**statistics**] [**detail**]

Displays 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:

```
Device# 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

Example: Specifying the Match Criteria for BGP Policy Accounting

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 30
set traffic-index 2
!
route-map set_bucket permit 20
match community 40
set traffic-index 3
!
route-map set_bucket permit 30
match community 50
set traffic-index 4
!
route-map set_bucket permit 40
match community 60
set traffic-index 5
```

Example: Classifying the IP Traffic and Enabling BGP Policy Accounting

In the following example, BGP policy accounting is enabled on POS interface 7/0 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 POS7/0
 ip address 10.15.1.2 255.255.255.0
 no ip directed-broadcast
 bgp-policy accounting
 no keepalive
 crc 32
 clock source internal
```

Additional References

The following sections provide references related to BGP policy accounting.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Switching Command Reference
CEF and dCEF configuration information	<i>IP Switching Cisco Express Forwarding Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
CISCO-BGP-POLICY-ACCOUNTING-MIB	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

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 47: Feature Information for BGP Policy Accounting Output Interface Accounting

Feature Name	Releases	Feature Information
BGP Policy Accounting Output Interface Accounting	12.0(9)S 12.0(17)ST 12.3(4)T Cisco IOS XE Release 2.1	<p>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.</p> <p>The following commands were introduced or modified:</p> <ul style="list-style-type: none"> • bgp-policy • set traffic-index • show cef interface • show cef interface policy statistics



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 Release 12.0(27)S, 12.3(8)T, 12.2(25)S, and later releases, support was introduced for mixed EIGRP MPLS VPN network topologies that contain VPN and backdoor links.

- [Finding Feature Information, page 567](#)
- [Prerequisites for the BGP Cost Community Feature, page 567](#)
- [Restrictions for the BGP Cost Community Feature, page 568](#)
- [Information About the BGP Cost Community Feature, page 568](#)
- [How to Configure the BGP Cost Community Feature, page 571](#)
- [Configuration Examples for the BGP Cost Community Feature, page 574](#)
- [Additional References, page 575](#)
- [Feature Information for BGP Cost Community, page 576](#)

Finding Feature Information

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

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

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

The cost community is a nontransitive, 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 community ID number determines the preference for the path selection process. The path with the lowest cost community ID 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 community ID number, the path selection process will then prefer the path with the lowest cost number. 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 a 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 the “Influencing Route Preference in a Multi-Exit IGP Network” section.

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)

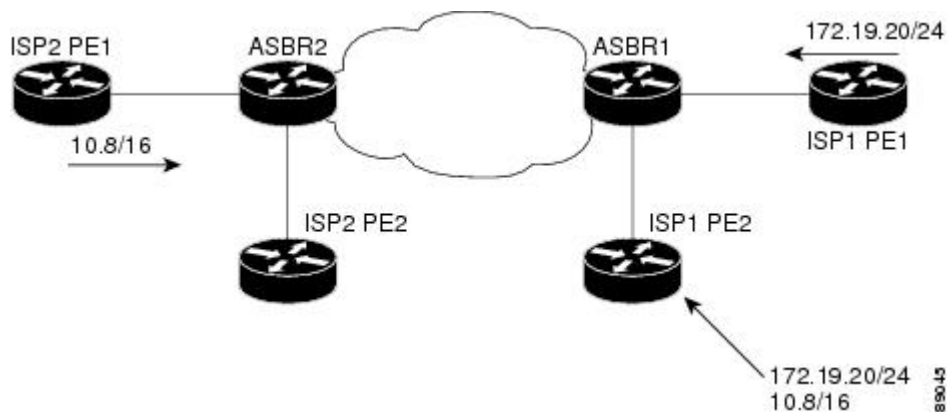
The single advertised path will include the aggregated cost communities as follows:

- {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 52: 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-bestpath" 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. This feature is enabled automatically for EIGRP VPN sites when Cisco IOS Release 12.0(27)S is installed to a PE, CE, or back door router.

For information about configuring EIGRP MPLS VPNs, refer to the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge document in Cisco IOS Release 12.0(27)S.

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) feature documentation in Cisco IOS Release 12.0(27)S.

How to Configure the BGP Cost Community Feature

Configuring the BGP Cost Community

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**] | **vpn4** [**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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 10.0.0.1 remote-as 101	Establishes peering with the specified neighbor or peer-group.
Step 5	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] ipv6 [multicast unicast] vpnv4 [unicast] Example: Device(config-router)# address-family ipv4	Places the router in address family configuration mode.
Step 6	neighbor <i>ip-address</i> route-map <i>map-name</i> {in out} Example: Device(config-router)# neighbor 10.0.0.1 route-map MAP-NAME in	Applies an incoming or outgoing route map for the specified neighbor or peer-group.
Step 7	exit Example: Device(config-router)# exit	Exits router configuration mode and enters global configuration mode.
Step 8	route-map <i>map-name</i> {permit deny} [<i>sequence-number</i>] Example: Device(config)# route-map MAP-NAME permit 10	Enters route map configuration mode to create or configure a route map.
Step 9	set extcommunity cost [igp] <i>community-id</i> <i>cost-value</i>	Creates a set clause to apply the cost community attribute.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-route-map)# set extcommunity cost 1 100</pre>	<ul style="list-style-type: none"> 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 <i>cost-value</i> 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 <i>cost-value</i>, which is half of the maximum value (4294967295) or 2147483647.
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	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 “Example: BGP Cost Community Verification” section for sample output.

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 is received from an external peer.

Configuration Examples for the BGP Cost Community Feature

Example: BGP Cost Community Configuration

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.

```
Device(config)# router bgp 50000
Device(config-router)# neighbor 10.0.0.1 remote-as 50000
Device(config-router)# neighbor 10.0.0.1 update-source Loopback 0
Device(config-router)# address-family ipv4
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 route-map COST1 in
Device(config-router-af)# neighbor 10.0.0.1 send-community both
Router(config-router-af)# exit
Device(config)# route-map COST1 permit 10
Device(config-route-map)# match ip-address 1
Device(config-route-map)# set extcommunity cost 1 100
```

Example: BGP Cost Community Verification

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:

```
Device# 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:

```
Device# show route-map

route-map set-cost, permit, sequence 10
  Match clauses:
  Set clauses:
```



```

extended community RT:1:1 RT:2:2 RT:3:3 RT:4:4 RT:5:5 RT:6:6 RT:7:7
  RT:100:100 RT:200:200 RT:300:300 RT:400:400 RT:500:500 RT:600:600
  RT:700:700 additive
extended community Cost:igp:1:4294967295 (default+2147483648)
  Cost:igp:2:200 Cost:igp:3:300 Cost:igp:4:400
  Cost:igp:5:2147483648 (default+1) Cost:igp:6:2147484648 (default+1001)
  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:

```
Device# 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:

```
Device# 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.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature	“EIGRP MPLS VPN PE-CE Site of Origin (SoO)” module in the <i>IP Routing: EIGRP Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
draft-retana-bgp-custom-decision-00.txt	BGP Custom Decision Process

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 48: Feature Information for BGP Cost Community

Feature Name	Releases	Feature Information
BGP Cost Community	12.0(24)S 12.3(2)T 12.2(18)S 12.2(27)SBC 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified: bgp bestpath cost-community ignore, debug ip bgp updates, and set extcommunity cost.</p>

Feature Name	Releases	Feature Information
BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links	12.0(27)S 12.3(8)T 12.2(25)S	<p>Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. The "pre-bestpath" 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 Release 12.0(27)S, 12.3(8)T, 12.2(25)S or later releases, is installed to a PE, CE, or back door router.</p> <p>No commands were introduced or modified.</p>



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 579](#)
- [Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 580](#)
- [Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 580](#)
- [Information About BGP Support for IP Prefix Import from Global Table into a VRF Table, page 580](#)
- [How to Import IP Prefixes from Global Table into a VRF Table, page 581](#)
- [Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 587](#)
- [Additional References, page 589](#)
- [Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 590](#)

Finding Feature Information

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

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

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.
- The global prefixes should be in the BGP table, so that this feature can import them into the BGP VRF table.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a second VPNv4 VRF.

Information About BGP Support for IP Prefix Import from Global Table into a VRF Table

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

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/length** | **permit network/length**} [**ge ge-value**] [**le le-value**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log] Example: Device(config)# access-list 50 permit 10.1.1.0 0.0.0.255	Creates an access list and defines a range of IP prefixes to import into the VRF table. <ul style="list-style-type: none"> • 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.
Step 4	ip prefix-list <i>prefix-list-name</i> [seq <i>seq-value</i>] {deny <i>network/length</i> permit <i>network/length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: Device(config)# ip prefix-list COLORADO permit 10.24.240.0/22	Creates a prefix list and defines a range of IP prefixes to import into the VRF table. <ul style="list-style-type: none"> • 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.

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. By default, a maximum of 1000 prefixes per VRF can be imported. You can change the limit to be from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you increase the prefix import limit above 1000. 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-AFIMPORT_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.



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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i>	Creates a VRF routing table and specifies the VRF name (or tag).

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config)# ip vrf GREEN</pre>	<ul style="list-style-type: none"> The ip vrf <i>vrf-name</i> command creates a VRF routing table and a CEF table, and both are named using the <i>vrf-name</i> argument. Associated with these tables is the default route distinguisher value.
Step 4	<p>rd <i>route-distinguisher</i></p> <p>Example:</p> <pre>Router(config-vrf)# rd 100:10</pre>	<p>Creates routing and forwarding tables for the VRF instance.</p> <ul style="list-style-type: none"> 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).
Step 5	<p>import ipv4 {unicast multicast} [<i>prefix-limit</i>] map <i>route-map</i></p> <p>Example:</p> <pre>Router(config-vrf)# import ipv4 unicast 1000 map UNICAST</pre>	<p>Imports IPv4 prefixes from the global routing table to a VRF table, filtered by the specified route map.</p> <ul style="list-style-type: none"> Unicast or multicast prefixes are specified. Up to a 1000 prefixes will be imported by default. The <i>prefix-limit</i> argument is used to specify a limit from 1 to 2,147,483,647 prefixes. The example references a route map that will import up to 1000 unicast prefixes that pass through the route map.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-vrf)# exit</pre>	<p>Exits VRF configuration mode and enters global configuration mode.</p>
Step 7	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map UNICAST permit 10</pre>	<p>Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.</p> <ul style="list-style-type: none"> The route map name must match the route map specified in Step 5. The example creates a route map named UNICAST.
Step 8	<p>match ip address {<i>acl-number</i> [<i>acl-number</i> <i>acl-name</i>] <i>acl-name</i> [<i>acl-name</i> <i>acl-number</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name</i>]}</p> <p>Example:</p> <pre>Router(config-route-map)# match ip address 50</pre>	<p>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.</p> <ul style="list-style-type: none"> Both IP access lists and IP prefix lists are supported. The example configures the route map to use standard access list 50 to define match criteria.
Step 9	<p>end</p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route-map configuration mode and returns to privileged EXEC mode.</p>

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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface Ethernet0/0	Configures an interface and enters interface configuration mode.
Step 4	ip policy route-map <i>map-tag</i> Example: Router(config-if)# ip policy route-map UNICAST	Identifies a route map to use for policy routing on an interface. <ul style="list-style-type: none"> • The example attaches the route map named UNICAST to the interface.

	Command or Action	Purpose
Step 5	ip verify unicast vrf <i>vrf-name</i> {deny permit} Example: <pre>Router(config-if)# ip verify unicast vrf GREEN permit</pre>	(Optional) Enables Unicast Reverse Path Forwarding verification for the specified VRF. <ul style="list-style-type: none"> The example enables verification for the VRF named GREEN. Traffic that passes verification will be forwarded.
Step 6	end Example: <pre>Router(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

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

- enable**
- show ip bgp vpnv4 {all | rd *route-distinguisher* | vrf *vrf-name*}**
- show ip vrf [brief | detail | interfaces | id] [*vrf-name*]**

DETAILED STEPS

Step 1	enable Enables privileged EXEC mode. Enter your password if prompted. Example: <pre>Device# enable</pre>
Step 2	show ip bgp vpnv4 {all rd <i>route-distinguisher</i> vrf <i>vrf-name</i>} 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: <pre>Device# show ip bgp vpnv4 all</pre> <p>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,</p>

```

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 `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:

```

Device# 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

Example: Importing IP Prefixes from Global Table into a VRF Table

The following example imports unicast prefixes into the VRF named *green* by using an IP prefix list and a route map:

This example starts in global configuration mode:

```

!
ip prefix-list COLORADO seq 5 permit 10.131.64.0/19
ip prefix-list COLORADO seq 10 permit 172.31.2.0/30
ip prefix-list COLORADO seq 15 permit 172.31.1.1/32
!
ip vrf green
 rd 200:1
  import ipv4 unicast map UNICAST
  route-target export 200:10
  route-target import 200:10
!
exit
!
route-map UNICAST permit 10
 match ip address prefix-list COLORADO
!
exit

```

Example: Verifying IP Prefix Import to a VRF Table

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 sample output shows that the import route map named UNICAST is importing IPv4 unicast prefixes and the prefix import limit is 1000:

```

Device# 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
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 sample output displays the import route map names, the prefix import limit and the actual number of imported prefixes, and the individual import entries:

```

Device# 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

```

```

*>i10.131.64.0/19 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
*>i172.31.1.1/32 10.131.95.252 0 100 0 i
*>i172.31.2.0/30 10.131.95.252 0 100 0 i
Route Distinguisher: 200:2 (default for vrf red)
*> 172.16.1.1/32 172.16.2.1 0 32768 i
*> 172.16.2.0/30 0.0.0.0 0 32768 i
*>i172.31.1.1/32 10.131.95.252 0 100 0 i
*>i172.31.2.0/30 10.131.95.252 0 100 0 i

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
MPLS Layer 3 VPN configuration tasks	“Configuring MPLS Layer 3 VPNs” module in the <i>MPLS: Layer 3 VPNs Configuration Guide</i>
VRF selection using policy based routing	“MPLS VPN VRF Selection Using Policy-Based Routing” module in the <i>MPLS: Layer 3 VPNs Configuration Guide</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

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

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 . An account on Cisco.com is not required.

Table 49: Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table

Feature Name	Releases	Feature Information
BGP Support for IP Prefix Import from Global Table into a VRF Table	12.0(29)S 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(14)T 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified by this feature: debug ip bgp import, import ipv4, and ip verify unicast vrf.</p>



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

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

- [Finding Feature Information, page 593](#)
- [Information About IP Prefix Export from a VRF Table into the Global Table, page 594](#)
- [How to Export IP Prefixes from a VRF Table into the Global Table, page 595](#)
- [Configuration Examples for IP Prefix Export from a VRF Table into the Global Table, page 602](#)
- [Additional References, page 603](#)
- [Feature Information for IP Prefix Export from a VRF Table into the Global Table, page 603](#)

Finding Feature Information

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

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

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

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

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

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

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

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

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



Caution

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

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

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

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

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

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

- **match as-path**

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



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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Device(config)# vrf definition vpn1	Creates a VRF routing table and specifies the VRF name (or tag).
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:100	Creates routing and forwarding tables for the VRF instance. <ul style="list-style-type: none"> • There are two formats for configuring the argument. It can be configured in the <i>as-number:network number (ASN:nn)</i> format, as shown in the example, or it can be configured in the <i>IP address:network number (IP-address:nn)</i>.

	Command or Action	Purpose
Step 5	address-family {ipv4 ipv6} Example: Device(config-vrf)# address-family ipv4	Configures the IPv4 or IPv6 address family.
Step 6	export {ipv4 ipv6} {unicast multicast} [prefix-limit] map map-name Example: Device(config-vrf-af)# export ipv4 unicast 500 map UNICAST	Exports IPv4 or IPv6 prefixes from the VRF table to the global routing table, filtered by the specified route map. <ul style="list-style-type: none"> • Specify ipv4 or ipv6, which you specified in Step 5. This example exports IPv4 unicast prefixes. • Based on this example, no more than 500 prefixes will be exported. • The prefixes exported are those that pass the route map.
Step 7	route-target import route-target-ext-community Example: Device(config-vrf-af)# route-target import 100:100	Creates a route-target extended community for a VRF instance. <ul style="list-style-type: none"> • For information about route-target import or export, see the <i>MPLS: Layer 3 VPNs Configuration Guide</i>.
Step 8	route-target export route-target-ext-community Example: Device(config-vrf-af)# route-target export 100:100	Creates a route-target extended community for a VRF instance.
Step 9	exit Example: Device(config-vrf-af)# exit	Exits address family configuration mode and enters global configuration mode.
Step 10	exit Example: Device(config-vrf)# exit	Exits VRF configuration mode and enters global configuration mode.
Step 11	route-map map-tag [permit deny] [sequence-number] Example: Device(config)# route-map UNICAST permit 10	Enables policy routing. <ul style="list-style-type: none"> • The example creates a route map named UNICAST.

	Command or Action	Purpose
Step 12	<p>match ip address {<i>acl-number</i> [<i>acl-number</i> <i>acl-name</i>] <i>acl-name</i> [<i>acl-name</i> <i>acl-number</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 50</pre>	<p>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.</p> <ul style="list-style-type: none"> • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria. • Define the access list (not shown in this task); for example, access-list 50 permit 192.168.1.0 255.255.255.0.
Step 13	<p>end</p> <p>Example:</p> <pre>Device(config-route-map)# end</pre>	<p>Exits route-map configuration mode and returns to privileged EXEC mode.</p>

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

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



Note

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i> Example: Device(config)# ip vrf GREEN	Creates a VRF routing table and specifies the VRF name (or tag). • The ip vrf <i>vrf-name</i> command creates a VRF routing table and a CEF table, and both are named using the <i>vrf-name</i> argument. Associated with these tables is the default route distinguisher value.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:10	Creates routing and forwarding tables for the VRF instance. • There are two formats for configuring the argument. It can be configured in the <i>as-number:network number (ASN:nn)</i> format, as shown in the example, or it can be configured in the <i>IP-address:network number (IP-address:nn)</i> .
Step 5	export ipv4 {unicast multicast} [<i>prefix-limit</i>] map <i>map-tag</i>	Exports IPv4 prefixes from the VRF table to the global routing table, filtered by the specified route map.

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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

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

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

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

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

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

Additional References

Related Documents

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Prefix Export from a VRF Table into the Global Table

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/cisco/web/featurenavigator](#). An account on Cisco.com is not required.

Table 50: Feature Information for BGP Support for IP Prefix Export from a VRF Table into the Global Table

Feature Name	Releases	Feature Information
BGP Support for IP Prefix Export from a VRF Table into the Global Table	15.1(2)SY	<p>This feature allows a network administrator to export IP prefixes from a VRF routing table into the global routing table.</p> <p>The following command was introduced: export map (VRF table to global table).</p> <p>The following commands were modified: debug ip bgp import and show ip bgp.</p>



CHAPTER 34

BGP per Neighbor SoO Configuration

The BGP per Neighbor SoO Configuration feature simplifies the configuration of the site-of-origin (SoO) value. In Cisco IOS Release 12.4(9)T, 12.2(33)SRA, 12.2(31)SB2, and previous releases, the SoO value is configured using an inbound route map that sets the SoO value during the update process. Per neighbor SoO configuration introduces two new commands that can be configured in submodes under router configuration mode to set the SoO value. In Cisco IOS Release 12.4(24)T, support was added for 4-byte autonomous system numbers in asdot format only.

- [Finding Feature Information, page 605](#)
- [Prerequisites for BGP per Neighbor SoO Configuration, page 605](#)
- [Restrictions for BGP per Neighbor SoO Configuration, page 606](#)
- [Information About Configuring BGP per Neighbor SoO, page 606](#)
- [How to Configure BGP per Neighbor SoO, page 608](#)
- [Configuration Examples for BGP per Neighbor SoO Configuration, page 619](#)
- [Additional References, page 621](#)
- [Feature Information for BGP per Neighbor SoO Configuration, page 622](#)

Finding Feature Information

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

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

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

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.

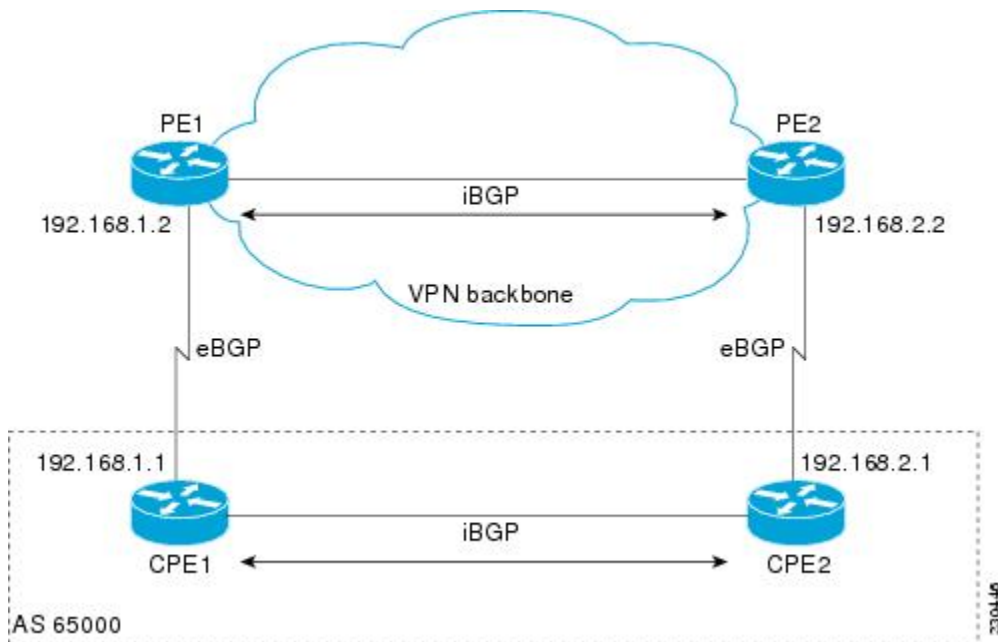
**Note**

A BGP neighbor or peer policy template-based SoO configuration takes precedence over the SoO value configured in an inbound route map.

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 53: Network Diagram for SoO Example



Benefits of BGP per Neighbor Site of Origin

In releases prior to the introduction of this feature, the SoO extended community attribute is configured using an inbound route map that sets the SoO value during the update process. With the introduction of the BGP per Neighbor Site of Origin feature, two new commands configured in submodes under router configuration mode simplify the SoO value configuration.

How to Configure BGP per Neighbor SoO

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ip cef Example: Device(config)# ip cef	Enables Cisco Express Forwarding on the route processor.
Step 4	ip vrf vrf-name Example: Device(config)# ip vrf SOO_VRF	Defines a VRF instance and enters VRF configuration mode.
Step 5	rd route-distinguisher Example: Device(config-vrf)# rd 1:1	Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN. <ul style="list-style-type: none"> • Use the <i>route-distinguisher</i> argument to specify the default RD for a VPN. There are two formats that you can use to specify an RD: <ul style="list-style-type: none"> • 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} <i>route-target-ext-community</i> Example: Device(config-vrf)# route-target export 1:1	Creates a route-target extended community for a VRF. <ul style="list-style-type: none"> • 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 <i>route-target-ext-community</i> argument to specify the VPN extended community. <p>Note Only the syntax applicable to this step is displayed. For a different use of this syntax, see Step 7.</p>
Step 7	route-target {import both} <i>route-target-ext-community</i> Example: Device(config-vrf)# route-target import 1:1	Creates a route-target extended community for a VRF. <ul style="list-style-type: none"> • Use the import keyword to import routing information from 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 <i>route-target-ext-community</i> argument to specify the VPN extended community.

	Command or Action	Purpose
Step 8	exit Example: Device(config-vrf)# exit	Exits VRF configuration mode and returns to global configuration mode.
Step 9	interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 1/0/0	Configures an interface type and enters interface configuration mode.
Step 10	ip vrf forwarding <i>vrf-name [downstream vrf-name2]</i> Example: Device(config-if)# ip vrf forwarding SOO_VRF	Associates a VRF with an interface or subinterface. <ul style="list-style-type: none"> In this example, the VRF named SOO_VRF is associated with Gigabit Ethernet interface 1/0/0. Note Executing this command on an interface removes the IP address, so the IP address should be reconfigured.
Step 11	ip address <i>ip-address mask [secondary]</i> Example: Device(config-if)# ip address 192.168.1.2 255.255.255.0	Configures an IP address. <ul style="list-style-type: none"> In this example, Gigabit Ethernet interface 1/0/0 is configured with an IP address of 192.168.1.2.
Step 12	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	show ip vrf [<i>brief detail interfaces id</i>] [<i>vrf-name</i>] [<i>output-modifiers</i>] Example: Device# show ip vrf	Displays the configured VRFs. <ul style="list-style-type: none"> Use this command to verify the configuration of this task.

Examples

The following output of the **show ip vrf** command displays the VRF named SOO_VRF configured in this task.

```
Device# 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.

For a configuration example involving 4-byte autonomous system numbers, see the “Example: Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template” section.

**Note**

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.

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.

Before You Begin

This task assumes that the task described in the "Verifying CEF and Configuring VRF Instances" section 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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 50000	Enters router configuration mode for the specified routing process.
Step 4	template peer-policy <i>policy-template-name</i> Example: Router(config-router)# template peer-policy SOO_POLICY	Creates a peer policy template and enters policy-template configuration mode.
Step 5	soo <i>extended-community-value</i> Example: Router(config-router-ptmp)# soo 65000:1	Sets the SoO value for a BGP peer policy template. • Use the <i>extended-community-value</i> argument to specify the VPN extended community value. The value takes one of the following formats: • A 16-bit autonomous system number, a colon, and a 32-bit number, for example: 45000:3

	Command or Action	Purpose
		<ul style="list-style-type: none"> • A 32-bit IP address, a colon, and a 16-bit number, for example: 192.168.10.2:51 • In this example, the SoO value is set at 65000:1.
Step 6	exit-peer-policy Example: <pre>Router(config-router-pmtp) # exit-peer-policy</pre>	Exits policy-template configuration mode and returns to router configuration mode.
Step 7	address-family ipv4 [unicast multicast] vrf vrf-name] Example: <pre>Router(config-router) # address-family ipv4 vrf SOO_VRF</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	neighbor ip-address remote-as autonomous-system-number Example: <pre>Router(config-router-af) # neighbor 192.168.1.1 remote-as 65000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 9	neighbor ip-address activate Example: <pre>Router(config-router-af) # neighbor 192.168.1.1 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.
Step 10	neighbor ip-address inherit peer-policy policy-template-name Example: <pre>Router(config-router-af) # neighbor 192.168.1.1 inherit peer-policy SOO_POLICY</pre>	<p>Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.</p> <ul style="list-style-type: none"> • In this example, the router is configured to send the peer policy template named SOO_POLICY to the 192.168.1.1 neighbor to inherit. If another peer policy template is indirectly inherited from SOO_POLICY, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from SOO_POLICY.

	Command or Action	Purpose
Step 11	end Example: Router(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.

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. For the IPv4 VRF address family, 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.

Before You Begin

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv4 [<i>unicast</i> <i>multicast</i> <i>vrf vrf-name</i>] Example: Device(config-router)# address-family ipv4 vrf SOO_VRF	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> Example: Device(config-router-af)# neighbor 192.168.2.1 remote-as 65000	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	neighbor <i>ip-address</i> activate Example: Device(config-router-af)# neighbor 192.168.2.1 activate	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router. <ul style="list-style-type: none"> • In this example, the external BGP peer at 192.168.2.1 is activated. <p>Note If a peer group has been configured in Step 5, do not use this step because BGP peer groups are activated when any parameter is configured. For example, a BGP peer group is activated when an SoO value is configured using the neighbor soo command in Step 7.</p>
Step 7	neighbor {<i>ip-address</i> <i>peer-group-name</i>} soo <i>extended-community-value</i>	Sets the site-of-origin (SoO) value for a BGP neighbor or peer group. <ul style="list-style-type: none"> • In this example, the neighbor at 192.168.2.1 is configured with an SoO value of 65000:1.

	Command or Action	Purpose
	Example: <pre>Device(config-router-af)# neighbor 192.168.2.1 soo 65000:1</pre>	
Step 8	end Example: <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

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.

Before You Begin

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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 50000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv4 [unicast multicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 vrf SOO_VRF	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • 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 <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 5	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor SOO_group peer-group</pre>	Creates a BGP peer group.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} soo <i>extended-community-value</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor SOO_group soo 65000:1</pre>	<p>Sets the site-of-origin (SoO) value for a BGP neighbor or peer group.</p> <ul style="list-style-type: none"> In this example, the BGP peer group, SOO_group, is configured with an SoO value of 65000:1.
Step 7	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 remote-as 65000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 8	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.
Step 9	<p>neighbor <i>ip-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 peer-group SOO_group</pre>	Assigns the IP address of a BGP neighbor to a peer group.
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP per Neighbor SoO Configuration

Example: Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template

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 the IPv4 VRF address family, 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
  template peer-policy SOO_POLICY
    soo 65000:1
  exit-peer-policy
  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
```

Example: Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template

The following example shows how to create a peer policy template and configure an SoO value using a 4-byte autonomous system number, 1.2 in asdot format, as part of the peer policy. Under the IPv4 VRF address family, a neighbor is identified and configured to inherit the peer policy that contains the SoO value.

```
router bgp 1.2
  template peer-policy SOO_POLICY
    soo 1.2:3
  exit-peer-policy
  address-family ipv4 vrf SOO_VRF
    neighbor 192.168.3.2 remote-as 1.14
    neighbor 192.168.3.2 activate
    neighbor 192.168.3.2 inherit peer-policy SOO_POLICY
  end
```

The following example shows how to create a peer policy template and configure an SoO value using a 4-byte autonomous system number, 65538 in asplain format, as part of the peer policy. Under the IPv4 VRF address family, a neighbor is identified and configured to inherit the peer policy that contains the SoO value.

```
router bgp 65538
  template peer-policy SOO_POLICY
    soo 65538:3
  exit-peer-policy
  address-family ipv4 vrf SOO_VRF
    neighbor 192.168.3.2 remote-as 65550
```

Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command

```
neighbor 192.168.3.2 activate
neighbor 192.168.3.2 inherit peer-policy SOO_POLICY
end
```

Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command

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 in the IPv4 VRF address family 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
neighbor 192.168.2.1 activate
neighbor 192.168.2.1 soo 65000:1
end
```

Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command and 4-Byte Autonomous System Numbers

The following example shows how to configure an SoO value for a BGP neighbor. In this example, all BGP neighbors, route targets, and SoO values use 4-byte autonomous system numbers in asplain format. After checking that CEF is enabled, a VRF instance named SOO_VRF is configured with route targets. In a BGP router session, a neighbor is configured in the IPv4 VRF address family, and an SoO value is configured for the neighbor.

```
show ip cef
ip vrf SOO_VRF
rd 100:200
route-target export 1.14:1
route-target import 1.14:1
exit
interface Ethernet 1/0
ip vrf forwarding SOO_VRF
ip address 192.168.2.2 255.255.255.0
exit
router bgp 1.2
address-family ipv4 vrf SOO_VRF
neighbor 192.168.2.1 remote-as 1.14
neighbor 192.168.2.1 activate
neighbor 192.168.2.1 soo 1.14:1
end
```

Example: Configuring a per Neighbor SoO Value Using a BGP Peer Group

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 in the IPv4 VRF address family, 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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
IP Switching commands	Cisco IOS IP Switching Command Reference

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP per Neighbor SoO Configuration

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 51: Feature Information for BGP per Neighbor SoO Configuration

Feature Name	Releases	Feature Information
BGP per Neighbor SoO Configuration	12.2(33)SB 12.2(33)SRB 12.4(1)T 15.0(1)SY	<p>The BGP per neighbor SOO configuration feature simplifies the configuration of the site-of-origin (SoO) parameter. In earlier releases, the SoO parameter is configured using an inbound route map that sets the SoO value during the update process. The per neighbor SoO configuration introduces two new commands that can be configured in submodes under router configuration mode to set the SoO value.</p> <p>The following commands were introduced by this feature: neighbor soo, and soo.</p>



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 623](#)
- [Prerequisites for Per-VRF Assignment of BGP Router ID, page 623](#)
- [Information About Per-VRF Assignment of BGP Router ID, page 624](#)
- [How to Configure Per-VRF Assignment of BGP Router ID, page 624](#)
- [Configuration Examples for Per-VRF Assignment of BGP Router ID, page 641](#)
- [Additional References, page 647](#)
- [Feature Information for Per-VRF Assignment of BGP Router ID, page 648](#)

Finding Feature Information

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

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

Prerequisites for Per-VRF Assignment of BGP Router ID

Before you configure this feature, Cisco Express Forwarding (CEF) or distributed CEF (dCEF) 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

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

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 of BGP Router ID 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

Configuring VRF Instances

Perform this task to configure VRF instances to be used with the Per-VRF Assignment of Router ID 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`.

Before You Begin

This task assumes that you have CEF or dCEF 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i> Example: Device(config)# ip vrf vrf_trans	Defines a VRF instance and enters VRF configuration mode.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 45000:2	Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN. <ul style="list-style-type: none"> • Use the <i>route-distinguisher</i> 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 “Route Distinguisher” section. • In this example, the RD uses an autonomous system number with the number 2 after the colon.
Step 5	route-target [import both] <i>route-target-ext-community</i>	Creates a route-target extended community for a VRF. <ul style="list-style-type: none"> • Use the import keyword to import routing information from the target VPN extended community.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-vrf)# route-target import 55000:5</pre>	<ul style="list-style-type: none"> Use the both keyword to both import routing information from and export routing information to the target VPN extended community. Use the <i>route-target-ext-community</i> argument to specify the VPN extended community.
Step 6	<p>route-target [export both] <i>route-target-ext-community</i></p> <p>Example:</p> <pre>Device(config-vrf)# route-target export 55000:1</pre>	<p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> 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 <i>route-target-ext-community</i> argument to specify the VPN extended community.
Step 7	<p>exit</p> <p>Example:</p> <pre>Device(config-vrf)# exit</pre>	Exits VRF configuration mode and returns to global configuration mode.
Step 8	Repeat Step 3 through Step 7 for each VRF to be defined.	

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.

Before You Begin

- This task assumes that you have CEF or dCEF enabled.
- This task assumes that VRF instances have been configured as shown in preceding “Configuring VRF Instances” task in this module.

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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface loopback0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> • In this example, loopback interface 0 is configured.
Step 4	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 172.16.1.1 255.255.255.255	Configures an IP address. <ul style="list-style-type: none"> • In this example, the loopback interface is configured with an IP address of 172.16.1.1.
Step 5	exit Example: Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	interface <i>type number</i> Example: Router(config)# interface serial2/0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> In this example, serial interface 2/0 is configured.
Step 7	ip vrf forwarding <i>vrf-name</i> [downstream <i>vrf-name2</i>] Example: Router(config-if)# ip vrf forwarding vrf_trans	Associates a VRF with an interface or subinterface. <ul style="list-style-type: none"> In this example, the VRF named vrf_trans is associated with serial interface 2/0. Note Executing this command on an interface removes the IP address. The IP address should be reconfigured.
Step 8	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 192.168.4.1 255.255.255.0	Configures an IP address. <ul style="list-style-type: none"> In this example, serial interface 2/0 is configured with an IP address of 192.168.4.1.
Step 9	Repeat Step 5 through Step 8 for each VRF to be associated with an interface.	--
Step 10	end Example: Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.
Step 11	show ip vrf [brief detail interfaces id] [<i>vrf-name</i>] Example: Router# show ip vrf interfaces	(Optional) Displays the set of defined VRFs and associated interfaces. <ul style="list-style-type: none"> In this example, the output from this command shows the VRFs that have been created and their associated interfaces.

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
```

```
Interface      IP-Address      VRF              Protocol
Serial2       192.168.4.1    vrf_trans        up
Serial3       192.168.5.1    vrf_user         up
```

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.

Before You Begin

This task assumes that you have previously created the VRF instances and associated them with interfaces. For more details, see the “Configuring VRF Instances” task and the “Associating VRF Instances with Interfaces” task earlier in this module.

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**] | **vpn4** [**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**] | **vpn4** [**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** [*tth*]
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 vpn4** {**all** | **rd** *route-distinguisher* | **vrf vrf-name**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 45000	Enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: Router(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process. <p>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>
Step 5	bgp log-neighbor-changes Example: Router(config-router)# bgp log-neighbor-changes	Enables logging of BGP neighbor resets.
Step 6	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.1.1 remote-as 45000	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. <ul style="list-style-type: none"> • If the <i>autonomous-system-number</i> argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor. • If the <i>autonomous-system-number</i> 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 7	neighbor {<i>ip-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i>	Allows BGP sessions to use any operational interface for TCP connections.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 update-source loopback0</pre>	<ul style="list-style-type: none"> 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 8	<p>address-family {ipv4 [mdt multicast unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast]}</p> <p>Example:</p> <pre>Router(config-router)# address-family vpn4</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p> <ul style="list-style-type: none"> The example creates a VPNv4 address family session.
Step 9	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 activate</pre>	<p>Activates the neighbor under the VPNv4 address family.</p> <ul style="list-style-type: none"> In this example, the neighbor 172.16.1.1 is activated.
Step 10	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>send-community {both standard extended}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 send-community extended</pre>	<p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1.
Step 11	<p>exit-address-family</p> <p>Example:</p> <pre>Router(config-router-af)# exit-address-family</pre>	<p>Exits address family configuration mode and returns to router configuration mode.</p>
Step 12	<p>address-family {ipv4 [mdt multicast unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast]}</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 vrf vrf_trans</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p> <ul style="list-style-type: none"> The example specifies that the VRF instance named <code>vrf_trans</code> is to be associated with subsequent IPv4 address family configuration commands.
Step 13	<p>redistribute connected</p> <p>Example:</p> <pre>Router(config-router-af)# redistribute connected</pre>	<p>Redistributes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> 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 <i>Cisco IOS IP Routing: BGP Command Reference</i>.

	Command or Action	Purpose
Step 14	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000</pre>	<p>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.</p> <ul style="list-style-type: none"> • If the <i>autonomous-system-number</i> argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor. • If the <i>autonomous-system-number</i> 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	<p>neighbor <i>ip-address</i> local-as <i>autonomous-system-number</i> [no-prepend [replace-as [<i>dual-as</i>]]]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend</pre>	<p>Customizes the AS_PATH attribute for routes received from an eBGP neighbor.</p> <ul style="list-style-type: none"> • 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	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} ebgp-multihop [<i>ttl</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2</pre>	<p>Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.</p> <ul style="list-style-type: none"> • 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.
Step 17	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 activate</pre>	<p>Activates the neighbor under the IPV4 address family.</p> <ul style="list-style-type: none"> • In this example, the neighbor 192.168.1.1 is activated.
Step 18	<p>neighbor <i>ip-address</i> allowas-in [<i>number</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1</pre>	<p>Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers.</p> <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
Step 19	no auto-summary Example: Router(config-router-af)# no auto-summary	Disables automatic summarization and sends subprefix routing information across classful network boundaries.
Step 20	no synchronization Example: Router(config-router-af)# no synchronization	Enables the Cisco IOS software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).
Step 21	bgp router-id {ip-address auto-assign} Example: Router(config-router-af)# bgp router-id 10.99.1.1	Configures a fixed router ID for the local BGP routing process. <ul style="list-style-type: none"> In this example, the specified BGP router ID is assigned for the VRF instance associated with this IPv4 address family configuration.
Step 22	Repeat Step 11 to Step 21 to configure another VRF instance.	--
Step 23	end Example: Router(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 24	show ip bgp vpnv4 {all rd route-distinguisher vrf vrf-name} Example: Router# show ip bgp vpnv4 all	(Optional) Displays VPN address information from the BGP table. <ul style="list-style-type: none"> 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 <i>Cisco IOS Multiprotocol Label Switching Command Reference</i> .

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 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.

Before You Begin

This task assumes that you have previously created the VRF instances as shown in the “Configuring VRF Instances” task in this module.

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] | vpnv4 [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] | vpnv4 [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}* **ebgp-multihop** [*tll*]
26. **neighbor** *{ip-address | peer-group-name}* **activate**
27. **neighbor** *ip-address* **allows-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 vpnv4** *{all | rd route-distinguisher | vrf vrf-name}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Router(config)# interface loopback0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> In this example, loopback interface 0 is configured.
Step 4	ip address ip-address mask [secondary] Example: Router(config-if)# ip address 172.16.1.1 255.255.255.255	Configures an IP address. <ul style="list-style-type: none"> In this example, the loopback interface is configured with an IP address of 172.16.1.1.
Step 5	exit Example: Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 6	interface type number Example: Router(config)# interface loopback1	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> In this example, loopback interface 1 is configured.
Step 7	ip vrf forwarding vrf-name [downstream vrf-name2] Example: Router(config-if)# ip vrf forwarding vrf_trans	Associates a VRF with an interface or subinterface. <ul style="list-style-type: none"> In this example, the VRF named vrf_trans is associated with loopback interface 1. <p>Note Executing this command on an interface removes the IP address. The IP address should be reconfigured.</p>
Step 8	ip address ip-address mask [secondary] Example: Router(config-if)# ip address 10.99.1.1 255.255.255.255	Configures an IP address. <ul style="list-style-type: none"> In this example, loopback interface 1 is configured with an IP address of 10.99.1.1.
Step 9	Repeat Step 5 through Step 8 for each VRF to be associated with an interface.	--

	Command or Action	Purpose
Step 10	exit Example: <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 11	router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode for the specified routing process.
Step 12	bgp router-id { <i>ip-address</i> vrf auto-assign } Example: <pre>Router(config-router)# bgp router-id vrf auto-assign</pre>	Configures a fixed router ID for the local BGP routing process. <ul style="list-style-type: none"> In this example, a BGP router ID is automatically assigned for each VRF instance.
Step 13	no bgp default ipv4-unicast Example: <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	Disables the IPv4 unicast address family for the BGP routing process. <p>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>
Step 14	bgp log-neighbor-changes Example: <pre>Router(config-router)# bgp log-neighbor-changes</pre>	Enables logging of BGP neighbor resets.
Step 15	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: <pre>Router(config-router)# neighbor 192.168.1.1 remote-as 45000</pre>	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. <ul style="list-style-type: none"> If the <i>autonomous-system-number</i> argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor. If the <i>autonomous-system-number</i> 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 { <i>ip-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i>	Allows BGP sessions to use any operational interface for TCP connections.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 update-source loopback0</pre>	<ul style="list-style-type: none"> 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	<p>address-family {ipv4 [mdt multicast unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast]}</p> <p>Example:</p> <pre>Router(config-router)# address-family vpn4</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p> <ul style="list-style-type: none"> The example creates a VPNv4 address family session.
Step 18	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 activate</pre>	<p>Activates the neighbor under the VPNv4 address family.</p> <ul style="list-style-type: none"> In this example, the neighbor 172.16.1.1 is activated.
Step 19	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>}</p> <p>send-community {both standard extended}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 send-community extended</pre>	<p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1.
Step 20	<p>exit-address-family</p> <p>Example:</p> <pre>Router(config-router-af)# exit-address-family</pre>	<p>Exits address family configuration mode and returns to router configuration mode.</p>
Step 21	<p>address-family {ipv4 [mdt multicast unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast]}</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 vrf vrf_trans</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p> <ul style="list-style-type: none"> The example specifies that the VRF instance named <code>vrf_trans</code> is to be associated with subsequent IPv4 address family configuration mode commands.
Step 22	<p>redistribute connected</p> <p>Example:</p> <pre>Router(config-router-af)# redistribute connected</pre>	<p>Redistributes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> 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 <i>Cisco IOS IP Routing: BGP Command Reference</i>.

	Command or Action	Purpose
Step 23	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000</pre>	<p>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.</p> <ul style="list-style-type: none"> • If the <i>autonomous-system-number</i> argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor. • If the <i>autonomous-system-number</i> 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	<p>neighbor <i>ip-address</i> local-as <i>autonomous-system-number</i> [no-prepend [replace-as [dual-as]]]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend</pre>	<p>Customizes the AS_PATH attribute for routes received from an eBGP neighbor.</p> <ul style="list-style-type: none"> • 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 25	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} ebgp-multihop [<i>tth</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2</pre>	<p>Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.</p> <ul style="list-style-type: none"> • 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.
Step 26	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 activate</pre>	<p>Activates the neighbor under the IPV4 address family.</p> <ul style="list-style-type: none"> • In this example, the neighbor 192.168.1.1 is activated.
Step 27	<p>neighbor <i>ip-address</i> allowas-in [<i>number</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1</pre>	<p>Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers.</p> <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
Step 28	no auto-summary Example: Router(config-router-af)# no auto-summary	Disables automatic summarization and sends subprefix routing information across classful network boundaries.
Step 29	no synchronization Example: Router(config-router-af)# no synchronization	Enables the Cisco IOS software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).
Step 30	Repeat Step 20 to Step 29 to configure another VRF instance.	--
Step 31	end Example: Router(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 32	show ip bgp vpnv4 {all rd route-distinguisher vrf vrf-name} Example: Router# show ip bgp vpnv4 all	(Optional) Displays VPN address information from the BGP table. <ul style="list-style-type: none"> 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 <i>Cisco IOS Multiprotocol Label Switching Command Reference</i> .

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 ?
r> 172.23.0.0      172.23.1.1           0           0 3 1 ?
*>i10.21.1.1/32    192.168.3.1          0          100         0 2 i
*> 10.52.1.0/24    172.23.1.1           0           0 3 1 ?
*> 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
*>i10.21.1.1/32    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 ?
```

```

*> 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
*>i10.52.1.0/24   192.168.3.1  0      100  0 ?
*>i10.52.2.1/32   192.168.3.1  0      100  0 3 i
*>i10.52.3.1/32   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

Example: Manually Configuring a BGP Router ID per VRF

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
!
interface Ethernet0/0
 ip vrf forwarding vrf_trans
 ip address 172.22.1.1 255.255.0.0
!
interface Ethernet1/0
 ip vrf forwarding vrf_user
 ip address 172.23.1.1 255.255.0.0
!
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

```

Example: Manually Configuring a BGP Router ID per VRF

```

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 vpv4 all** command shows the router ID displayed next to the VRF name:

```

Router# show ip bgp vpv4 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
*> 172.22.0.0                0.0.0.0          0           32768 ?
r> 172.23.0.0                172.23.1.1      0           0 3 1 ?
*>i10.21.1.1/32              192.168.3.1     0          100        0 2 i
*> 10.52.1.0/24              172.23.1.1      0           0 3 1 ?
*> 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 ?
*> 10.99.2.2/32              0.0.0.0         0           32768 ?
Route Distinguisher: 50000:1
*>i10.21.1.1/32              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           0 2 1 2 i
*>i10.52.1.0/24              192.168.3.1     0          100        0 ?
*>i10.52.2.1/32              192.168.3.1     0          100        0 3 i
*>i10.52.3.1/32              192.168.3.1     0          100        0 3 i
*> 10.99.1.1/32              0.0.0.0         0           32768 ?
*> 10.99.2.2/32              172.22.1.1      0           0 2 1 ?

```

The output of the **show ip bgp vpv4 vrf** command for a specified VRF displays the router ID in the output header:

```

Router# show ip bgp vpv4 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
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           0 2 1 2 i
*>i10.52.1.0/24              192.168.3.1     0          100        0 ?
*>i10.52.2.1/32              192.168.3.1     0          100        0 3 i
*>i10.52.3.1/32              192.168.3.1     0          100        0 3 i
*> 10.99.1.1/32              0.0.0.0         0           32768 ?
*> 10.99.2.2/32              172.22.1.1      0           0 2 1 ?

```

The output of the **show ip bgp vpv4 vrf summary** command for a specified VRF displays the router ID in the first line of the output:

```

Router# show ip bgp vpv4 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 route-map cache entries using 0 bytes of memory
0 BGP filter-list 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 vpv4 vrf** command for a specified VRF and network address:

```

Router# show ip bgp vpv4 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

```

Example: Automatically Assigning a BGP Router ID per VRF

The following three 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

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 vpv4** 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
!
interface Ethernet0/0
 ip vrf forwarding vrf_trans
 ip address 172.22.1.1 255.0.0.0
!
interface Ethernet1/0
 ip vrf forwarding vrf_user
 ip address 172.23.1.1 255.0.0.0

```

Example: Automatically Assigning a BGP Router ID per VRF

```

!
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 “Example: Manually Configuring a BGP Router ID per VRF” section.

```
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 ?
*>i10.21.1.1/32    192.168.3.1          0    100     0 2 i
*> 10.52.1.0/24    172.23.1.1          0           0 3 1 ?
*> 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: 50000:1
*>i10.21.1.1/32    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           0 2 1 2 i
*>i10.52.1.0/24    192.168.3.1          0    100     0 ?
*>i10.52.2.1/32    192.168.3.1          0    100     0 3 i
*>i10.52.3.1/32    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

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
!
interface Ethernet0/0
 ip vrf forwarding vpn1
 ip address 172.22.1.1 255.0.0.0
!
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 remote-as 40000
  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
Ethernet0/0	172.22.1.1	YES	NVRAM	up	up
Ethernet1/0	unassigned	YES	NVRAM	administratively down	down
Serial2/0	unassigned	YES	NVRAM	administratively down	down
Serial3/0	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
```

Name	Default RD	Interfaces
vpn1	45000:1	Loopback0 Ethernet0/0

```
VRF session is established:
```

Per-VRF Automatically Assigned Router ID

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 vpv4** 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
```

Example: Automatically Assigning a BGP Router ID per VRF

```

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
!
interface Ethernet0/0
ip vrf forwarding vrf_trans
ip address 172.22.1.1 255.0.0.0
!
interface Ethernet1/0
ip vrf forwarding vrf_user
ip address 172.23.1.1 255.0.0.0
!
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 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

```

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 ?
*>i10.21.1.1/32    192.168.3.1         0           100      0 2 i
*> 10.52.1.0/24    172.23.1.1          0             0 3 1 ?

```



```

*> 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
*>i10.21.1.1/32      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          0 2 1 2 i
*>i10.52.1.0/24      192.168.3.1              0 100      0 ?
*>i10.52.2.1/32      192.168.3.1              0 100      0 3 i
*>i10.52.3.1/32      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 ?

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

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

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 52: Feature Information for Per-VRF Assignment of BGP Router ID

Feature Name	Releases	Feature Information
Per-VRF Assignment of BGP Router ID	12.2(31)SB2 12.2(33)SRA 12.2(33)SXH 12.4(20)T 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified by this feature: bgp router-id, show ip bgp vpv4.</p>



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 651](#)
- [Information About BGP Next Hop Unchanged, page 651](#)
- [How to Configure BGP Next Hop Unchanged, page 652](#)
- [Configuration Example for BGP Next Hop Unchanged, page 654](#)
- [Additional References, page 655](#)
- [Feature Information for BGP Next Hop Unchanged, page 655](#)

Finding Feature Information

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

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

Information About BGP Next Hop Unchanged

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.

**Note**

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {*ipv4* | *ipv6* | *l2vpn* | *nsap* | *rtfilter* | *vpn4* | *vpn6*}
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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65535	Enters router configuration mode, and creates a BGP routing process.
Step 4	address-family {ipv4 ipv6 l2vpn nsap rtfilter vpnv4 vpnv6} Example: Router(config-router-af)# address-family vpnv4	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.
Step 5	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: Router(config-router-af)# neighbor 10.0.0.100 remote-as 65600	Adds an entry to the BGP neighbor table.
Step 6	neighbor <i>ip-address</i> activate Example: Router(config-router-af)# neighbor 10.0.0.100 activate	Enables the exchange of information with the peer.
Step 7	neighbor <i>ip-address</i> ebgp-multihop <i>ttl</i> Example: Router(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255	Configures the local router to accept and initiate connections to external peers that reside on networks that are not directly connected.

	Command or Action	Purpose
Step 8	neighbor ip-address next-hop-unchanged Example: <pre>Router(config-router-af)# neighbor 10.0.0.100 next-hop-unchanged</pre>	Configures the router to send BGP updates to the specified eBGP peer without modifying the next hop attribute.
Step 9	end Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode, and enters privileged EXEC mode.
Step 10	show ip bgp Example: <pre>Router# show ip bgp</pre>	(Optional) Displays entries in the BGP routing table. <ul style="list-style-type: none"> • The output will indicate if the neighbor next-hop-unchanged command has been configured for the selected address.

Configuration Example for BGP Next Hop Unchanged

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
```



Note

All address families, such as IPv4, IPv6, VPNv4, VPNv6, L2VPN, and so on support the **next-hop unchanged** command. However, for the address family L2VPN BGP VPLS signaling, you must use the **next-hop self** command for its proper functioning.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer	“Configuring Internal BGP Features” in the <i>IP Routing: BGP Configuration Guide</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/cisco/web/featurenavigator/index.html](#). An account on Cisco.com is not required.

Table 53: Feature Information for BGP Next Hop Unchanged

Feature Name	Releases	Feature Configuration Information
BGP Next Hop Unchanged	12.0(22)S 12.0(16)ST 12.2 12.2(14)S 15.0(1)S	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 introduced by this feature: neighbor next-hop-unchanged.



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 657](#)
- [Prerequisites for BGP Event-Based VPN Import, page 657](#)
- [Information About BGP Event-Based VPN Import, page 658](#)
- [How to Configure BGP Event-Based VPN Import, page 659](#)
- [Configuration Examples for BGP Event-Based VPN Import, page 665](#)
- [Additional References, page 666](#)
- [Feature Information for BGP Event-Based VPN Import, page 666](#)

Finding Feature Information

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

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

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

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

The BGP Event-Based VPN Import feature 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.
- Bestpath—Import the best available path that matches the RT of the VRF instance. If the bestpath 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 bestpath and all paths marked as multipaths that match the RT of the VRF instance. If there are no bestpath or multipath matches, then the best available path is selected.

Multipath and bestpath 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, the software disables the fall back safety option of choosing the best available path. If there are no paths appropriate to the configured option (bestpath or multipath) in the exporting net that 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 bestpath 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

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

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router> enable</pre>	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>vrf definition <i>vrf-name</i></p> <p>Example:</p> <pre>Router(config)# vrf definition vrf-A</pre>	<p>Configures a VRF routing table and enters VRF configuration mode.</p> <ul style="list-style-type: none"> • Use the <i>vrf-name</i> argument to specify a name to be assigned to the VRF.
Step 4	<p>rd <i>route-distinguisher</i></p> <p>Example:</p> <pre>Router(config-vrf)# rd 45000:1</pre>	<p>Creates routing and forwarding tables and specifies the default route distinguisher for a VPN.</p> <ul style="list-style-type: none"> • Use the <i>route-distinguisher</i> argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 5	<p>route-target {import export both} <i>route-target-ext-community</i></p> <p>Example:</p> <pre>Router(config-vrf)# route-target both 45000:100</pre>	<p>Creates a route target extended community for a VRF.</p> <ul style="list-style-type: none"> • Use the import keyword to import routing information from the target VPN extended community. • 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 <i>route-target-ext-community</i> 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.
Step 6	<p>address-family ipv4 [unicast]</p> <p>Example:</p> <pre>Router(config-vrf)# address-family ipv4 unicast</pre>	<p>Specifies the IPv4 address family and enters VRF address family configuration mode.</p> <ul style="list-style-type: none"> • This step is required here to specify an address family for the VRF defined in the previous steps.
Step 7	<p>exit-address-family</p> <p>Example:</p> <pre>Router(config-vrf-af)# exit-address-family</pre>	Exits VRF address family configuration mode and returns to VRF configuration mode.

	Command or Action	Purpose
Step 8	exit Example: <pre>Router(config-vrf)# exit</pre>	Exits VRF configuration mode and enters global configuration mode.
Step 9	interface <i>type number</i> Example: <pre>Router(config)# interface FastEthernet 1/1</pre>	Enters interface configuration mode.
Step 10	vrf forwarding <i>vrf-name</i> Example: <pre>Router(config-if)# vrf forwarding vrf-A</pre>	Associates a VRF instance with the interface configured in Step 9. <ul style="list-style-type: none"> • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 10.4.8.149 255.255.255.0</pre>	Configures an IP address for the interface.
Step 12	no shutdown Example: <pre>Router(config-if)# no shutdown</pre>	Restarts a disabled interface.
Step 13	exit Example: <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
Step 14	Repeat Step 3 through Step 13 to bind other VRF instances with an interface.	--
Step 15	end Example: <pre>Router(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.

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.

Before You Begin

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” section earlier in this module.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the “Example: Configuring Event-Based VPN Import Processing for BGP Paths” section in this module.

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

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 45000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>address-family ipv4 vrf vrf-name</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 vrf vrf-A</pre>	<p>Specifies the IPv4 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> Use the vrf keyword and <i>vrf-name</i> argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	<p>import path selection {all bestpath [strict] multipath [strict]}</p> <p>Example:</p> <pre>Router(config-router-af)# import path selection all</pre>	<p>Specifies the BGP path selection policy for importing routes into a VRF table.</p> <ul style="list-style-type: none"> In this example, all paths that match any RT of the VRF instance are imported.
Step 6	<p>import path limit number-of-import-paths</p> <p>Example:</p> <pre>Router(config-router-af)# import path limit 3</pre>	<p>Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.</p>
Step 7	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

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

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} [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

BBGP 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
    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
    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

Example: Configuring Event-Based VPN Import Processing for BGP Paths

In this example, 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.

```
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 192.168.3.2 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

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	<i>Route Refresh Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 54: Feature Information for BGP Event-Based VPN Import

Feature Name	Releases	Feature Information
BGP Event-Based VPN Import	15.0(1)SY	<p>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.</p> <p>The following commands were introduced or modified: bgp scan-time, import path limit, import path selection, maximum-paths eibgp, maximum-paths ibgp, show ip bgp vpnv4.</p>



CHAPTER 38

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 Interior Border Gateway Protocol (iBGP) session (of another border router) as the best path for a prefix even if it has an Exterior Border Gateway Protocol (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 devices advertise one externally learned path called the best external path.

- [Finding Feature Information, page 669](#)
- [Prerequisites for BGP Best External, page 670](#)
- [Restrictions for BGP Best External, page 670](#)
- [Information About BGP Best External, page 671](#)
- [How to Configure BGP Best External, page 673](#)
- [Configuration Examples for BGP Best External, page 678](#)
- [Additional References, page 679](#)
- [Feature Information for BGP Best External, page 680](#)

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 for BGP Best External, on page 680](#).

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://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 Virtual Routing and Forwarding Label
- The BGP Best External feature cannot be configured with Multicast or Layer 2 VPN (L2VPN) Virtual Routing and Forwarding address families.
- The BGP Best External feature cannot be configured on route reflectors.
- The BGP Best External feature does not support Nonstop Forwarding/Stateful Switchover (NSF/SSO). However, In-Service Software Upgrade (ISSU) is supported if both Route Processors have the BGP Best External feature configured.
- The BGP Best External feature can be configured only 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 Prefix-Independent Convergence (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 need not remove the **protection local-prefixes** command from the configuration.

Information About BGP Best External

BGP Best External Overview

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 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 devices 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 Multiprotocol Label Switching (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 Interior Border Gateway Protocol (iBGP) session between them.
- Two routers in different autonomous systems of a confederation that have an External Border Gateway Protocol (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.

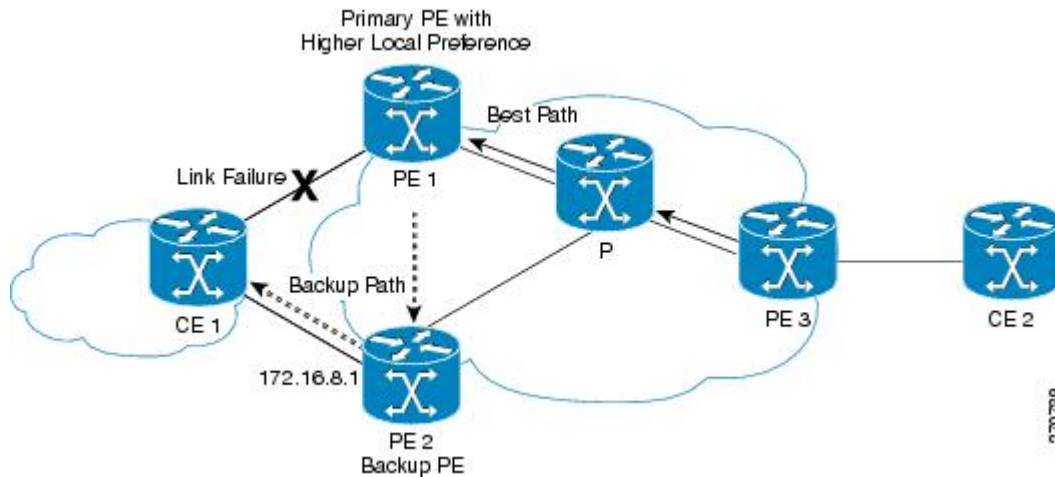
BGP Best External Feature Operation

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 Border Gateway Protocol (BGP) Prefix-Independent Convergence (PIC) functionality enables the best external path to take over, enabling faster restoration of connectivity.

[BGP Best External Feature Operation](#) shows a Multiprotocol Label Switching (MPLS) VPN using the BGP Best External feature. The network includes the following components:

Figure 54: MPLS VPN: Best External at the Edge of MPLS VPN



- Exterior BGP (eBGP) sessions exist between the provider edge (PE) and customer edge (CE) routers.
- PE 1 is the primary router and has a higher local preference setting.
- Traffic from CE 2 uses PE 1 to reach router CE 1.
- PE 1 has two paths to reach CE 1.
- CE 1 is dual-homed with PE 1 and PE 2.
- PE 1 is the primary path and PE 2 is the backup path.

In [BGP Best External Feature Operation](#), traffic in the MPLS cloud flows through PE 1 to reach CE 1. Therefore, PE 2 uses PE 1 as the best path and PE 2 as the backup path.

PE 1 and PE 2 are configured with the BGP Best External feature. BGP computes both the best path (the PE 1-CE 1 link) and a backup path (PE 2) and installs both paths into the Routing Information Base (RIB) and Cisco Express Forwarding. The best external path (PE 2) is advertised to the peer routers, in addition to the best path.

When Cisco Express Forwarding detects a link failure on the PE 1-CE 1 link, Cisco Express Forwarding immediately switches to the backup path PE 2. 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 Virtual Routing and Forwarding (VRF) 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 to that VRF.

How to Configure BGP Best External

Configuring the BGP Best External Feature

Perform the following task to configure the BGP Best External feature. This task shows how to configure the BGP Best External feature in either an IPv4 or VPNv4 address family. In VPNv4 address family configuration mode, the BGP Best External feature applies to all IPv4 Virtual Routing Forwarding (VRF); you need not 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 to that VRF.

Before You Begin

- Configure the MPLS VPN and verify that it is working properly before configuring the BGP Best External feature. See the "Configuring MPLS Layer 3 VPNs" section for more information.
- Configure multiprotocol VRFs to 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 the "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs section".
- Ensure that the customer edge (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**]
 - or
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** [**bfd** | **route-map** *map-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	Do one of the following: <ul style="list-style-type: none"> • address-family ipv4 [unicast vrf <i>vrf-name</i>] • or • address-family vpnv4 [unicast] • or Example: Router(config-router)# address-family ipv4 unicast Example: Router(config-router)# address-family vpnv4	Specifies the IPv4 or VPNv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 or VPNv4 unicast address family. • The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	bgp advertise-best-external Example: Router(config-router-af)# bgp advertise-best-external	Calculates and uses an external backup path and installs it into the RIB and Cisco Express Forwarding.

	Command or Action	Purpose
Step 6	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</p> <ul style="list-style-type: none"> 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	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.</p>
Step 8	<p>neighbor <i>ip-address</i> fall-over [bfd route-map <i>map-name</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd</pre>	<p>Configures the BGP peering to use fast session deactivation and enables BFD protocol support for failover.</p> <ul style="list-style-type: none"> BGP will remove all routes learned through this peer if the session is deactivated.
Step 9	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>(Optional) Exits address family configuration mode and returns to privileged EXEC mode.</p>

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 vpnv4 {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 bgp 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 vpnv4**{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-regex**] [**regex**] [**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)
*>i1.0.0.1/32     10.10.3.3           0       200     0 1 ?
* i              10.10.3.3           0       200     0 1 ?
*                10.0.0.1            0         0     0 1 ?
*bx             10.0.0.1            0         0     0 1 ?
*                10.0.0.1            0         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 10:10:10.10.10/32, 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
ia - 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

```

Configuration Examples for BGP Best External

Example: Configuring the BGP Best External Feature

The following example shows how to configure the BGP Best External feature in VPNv4 mode:

```

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
no synchronization
bgp recursion host
neighbor 192.168.13.2 remote-as 64511
neighbor 192.168.13.2 fall-over bfd
neighbor 192.168.13.2 activate
neighbor 192.168.13.2 as-override
exit-address-family

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Basic MPLS VPNs	“Configuring MPLS Layer 3 VPNs” module in the <i>MPLS: Layer 3 VPNs Configuration Guide</i>
Multiprotocol VRFs	“MPLS VPN VRF CLI for IPv4 and IPv6 VPNs” module in the <i>MPLS: Layer 3 VPNs Configuration Guide</i>
A failover feature that creates a new path after a link or node failure	MPLS VPN--BGP Local Convergence

Standards

Standard	Title
draft-marques-idr-best-external	<i>BGP Best External, Advertisement of the best external route to iBGP</i>

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1771	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 2547	<i>BGP/MPLS VPNs</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 55: Feature Information for BGP Best External

Feature Name	Releases	Feature Information
BGP Best External	12.2(33)SRE 15.2(3)T 15.2(4)S 15.1(1)SY	<p>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.</p> <p>The following commands were introduced or modified: bgp advertise-best-external, bgp recursion host, show ip bgp, show ip bgp vpnv4, show ip cef, show ip cef vrf, show ip route, and show ip route vrf.</p>



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 683](#)
- [Prerequisites for BGP PIC, page 684](#)
- [Restrictions for BGP PIC, page 684](#)
- [Information About BGP PIC, page 684](#)
- [How to Configure BGP PIC, page 694](#)
- [Configuration Examples for BGP PIC, page 697](#)
- [Additional References, page 700](#)
- [Feature Information for BGP PIC, page 701](#)

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 module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://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.
- Ensure that 2KB memory is available per prefix on the RP, SP and each line card. For example, if you need to scale up to 100000 prefixes, ensure that at least 200 MB is free on the RP, SP and each line card.

Restrictions for BGP PIC

- 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

- 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 an 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.

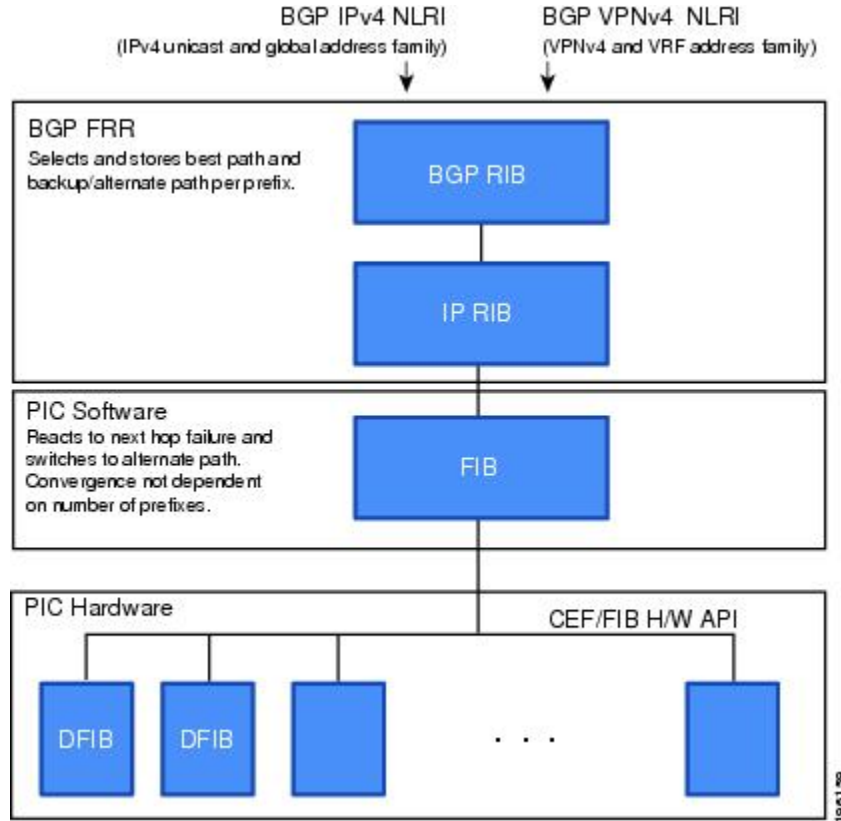
BGP Fast Reroute's Role in the BGP PIC Feature

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 55: BGP PIC Edge and BGP FRR



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 Achieves Subsecond Convergence

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:

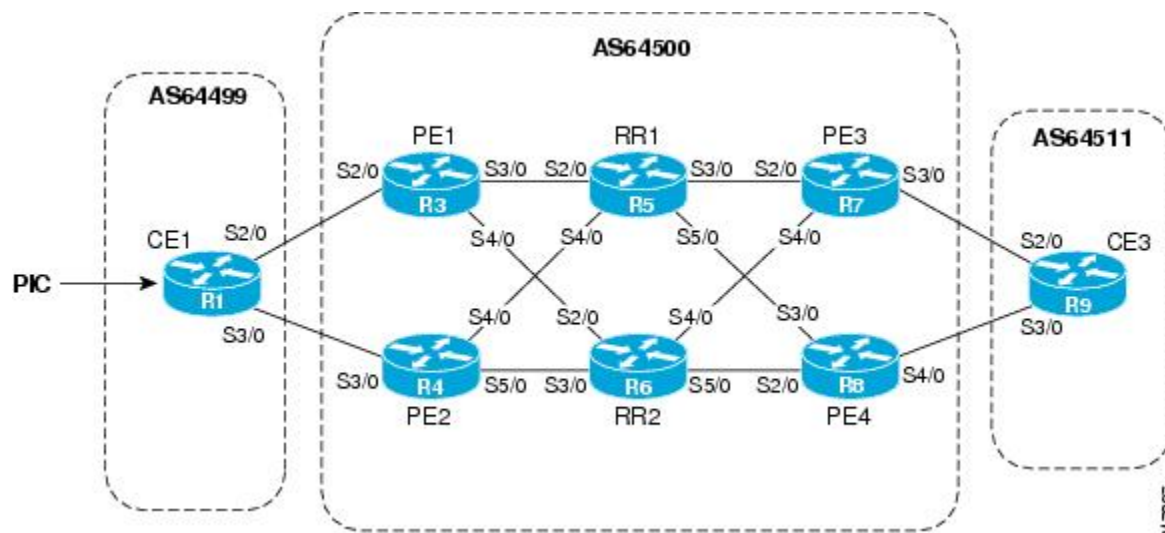
IP PE-CE Link and Node Protection on the CE Side (Dual PEs)

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 56: 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.

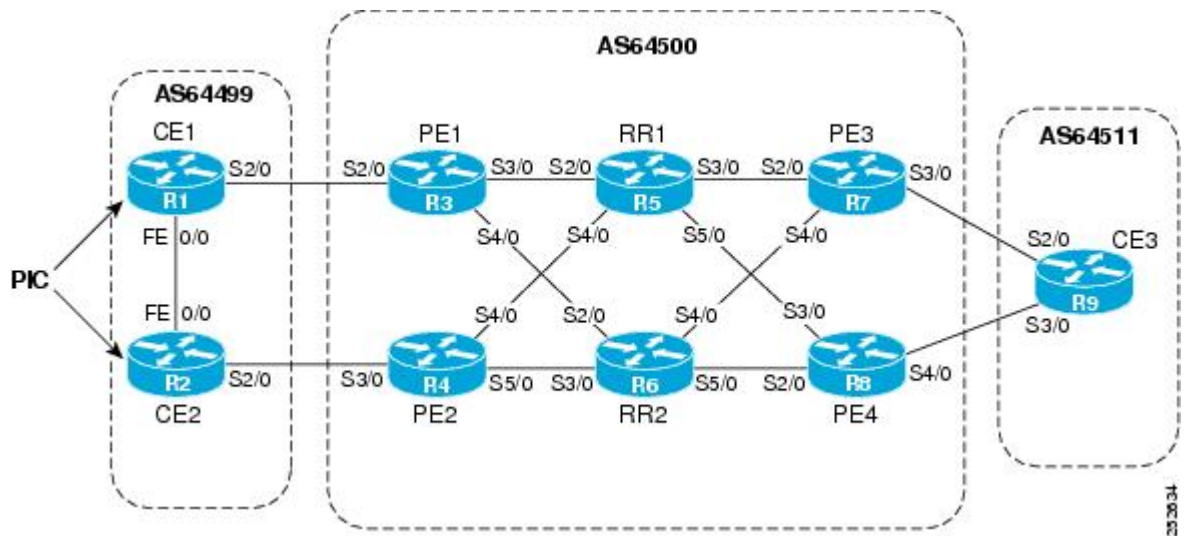
There should not be any policies set on CE1 and CE2 for the eBGP peers PE1 and PE2. Both CE routers must point to the eBGP route as next hop. On CE1, the next hop to reach CE3 is through PE1, so PE1 is the best path to reach CE3. On CE2, the best path to reach CE3 is PE2. CE2 advertises itself as the 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.

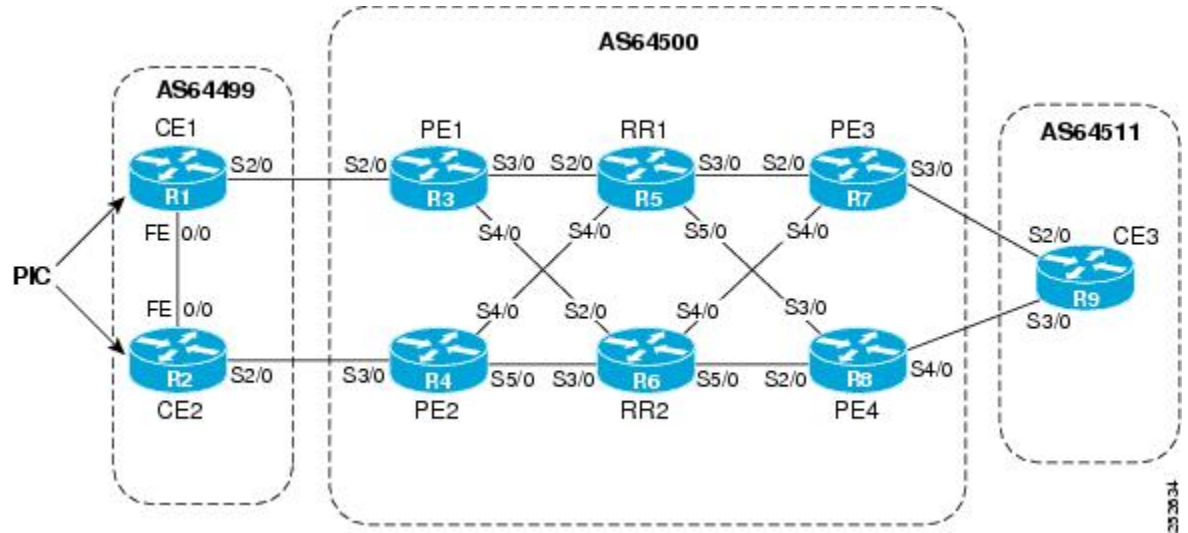
Figure 57: Using BGP PIC in a Dual CE, Dual PE Network



IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path

The figure below shows a network that uses the BGP PIC feature on CE1 and CE2. The network includes the following components:

Figure 58: Using BGP PIC in a Dual CE, Dual PE Network



- 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 the 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.

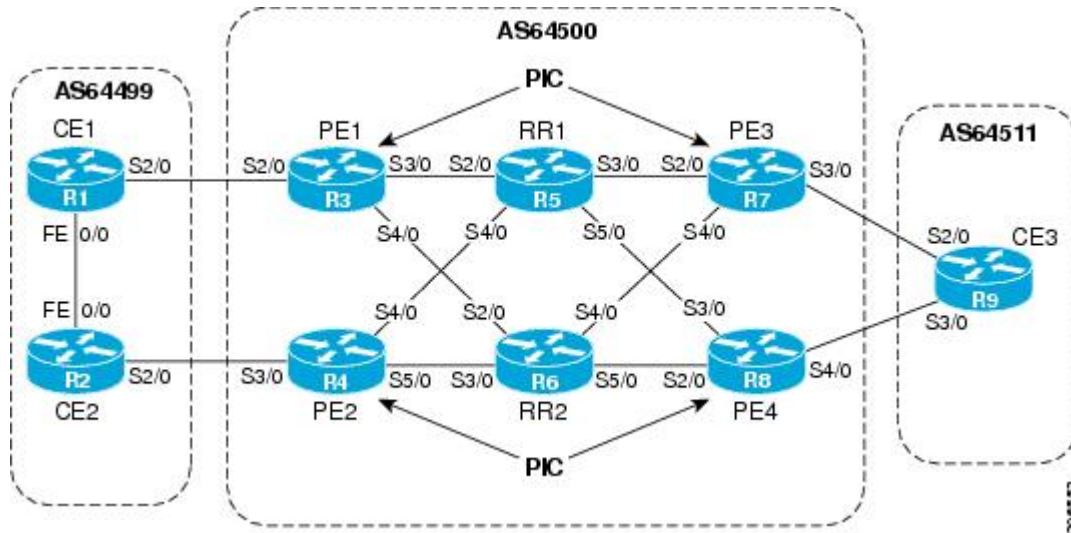
Thus, with BGP PIC enabled on PE3 and PE4, Cisco Express Forwarding detects the link failure and does in-place modification of the forwarding object to the backup/alternate node PE4 that already exists in Cisco

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 59: Enabling BGP PIC on All PE 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 the 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.



Note

When the BGP PIC feature is enabled, by default, **bgp recursion host** is configured for VPNv4 and VPNv6 address families and disabled for IPv4 and IPv6 address families.

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

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.

Before You Begin

- 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 the *MPLS: Layer 3 VPNs Configuration Guide* 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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	Do one of the following: <ul style="list-style-type: none"> • address-family ipv4 [unicast vrf <i>vrf-name</i>] • or • address-family vpnv4 [unicast] 	Specifies the IPv4 or VPNv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 or VPNv4 unicast address family. • The vrf keyword and <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre> <p>Example:</p> <pre>Device(config-router)# address-family vpv4</pre>	with subsequent IPv4 address family configuration mode commands.
Step 5	<p>bgp additional-paths install</p> <p>Example:</p> <pre>Device(config-router-af)# bgp additional-paths install</pre>	Calculates a backup/alternate path and installs it into the RIB and Cisco Express Forwarding.
Step 6	<p>neighbor ip-address remote-as autonomous-system-number</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</p> <ul style="list-style-type: none"> • 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	<p>neighbor ip-address activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 activate</pre>	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 8	<p>bgp recursion host</p> <p>Example:</p> <pre>Device(config-router-af)# bgp recursion host</pre>	<p>(Optional) Enables the recursive-via-host flag for IPv4, VPNv4, and VRF address families.</p> <ul style="list-style-type: none"> • 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	<p>neighbor ip-address fall-over [bfd route-map map-name]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.1 fall-over bfd</pre>	Enables BFD protocol support to detect when a neighbor has gone away, which can occur within a subsecond.

	Command or Action	Purpose
Step 10	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP PIC

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.

```
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
!

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
```

Example: Displaying Backup Alternate Paths for BGP PIC

```

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:

```

Device# 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
      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, 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:11:12 , recursive-via-connected

```

The command output in the following example shows that the VRFs in VRF green have backup/alternate paths:

```

Device# 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 11:12:12.0.0.0/24
    10.11.11.11 (via blue) from 10.11.11.11 (1.0.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:11:12 , recursive-via-connected
  1
    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
    10.13.13.13 from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
      Extended Community: RT:12:23 , recursive-via-connected
  1
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      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

```

The command output in the following example shows the BGP routing table entries for the backup and alternate paths:

```

Device# 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      3
  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:

```

Device# 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:

```
Device# 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
  recursive via 10.0.101.1, repair
    attached to
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
A failover feature that creates a new path after a link or node failure	MPLS VPN--BGP Local Convergence
Configuring multiprotocol VRFs	MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1771	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 2547	<i>BGP/MPLS VPNs</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 56: Feature Information for BGP PIC

Feature Name	Releases	Feature Information
BGP PIC Edge for IP and MPLS-VPN	15.1(1)SY	<p>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.</p> <p>The following commands were introduced or modified: bgp additional-paths install, bgp recursion host, show ip bgp, show ip cef, show ip route, and show vrf.</p>



Configuring BGP: RT Constrained Route Distribution

BGP: RT Constrained Route Distribution is a feature that can be used by service providers in Multiprotocol Label Switching (MPLS) Layer 3 VPNs to reduce the number of unnecessary routing updates that route reflectors (RRs) send to Provider Edge (PE) routers. The reduction in routing updates saves resources by allowing RRs, Autonomous System Boundary Routers (ASBRs), and PEs to have fewer routes to carry. Route targets are used to constrain routing updates.

- [Finding Feature Information, page 703](#)
- [Prerequisites for BGP: RT Constrained Route Distribution, page 704](#)
- [Restrictions for BGP: RT Constrained Route Distribution, page 704](#)
- [Information About BGP: RT Constrained Route Distribution, page 704](#)
- [How to Configure RT Constrained Route Distribution, page 708](#)
- [Configuration Examples for BGP: RT Constrained Route Distribution, page 719](#)
- [Additional References, page 720](#)
- [Feature Information for BGP: RT Constrained Route Distribution, page 722](#)

Finding Feature Information

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

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

Prerequisites for BGP: RT Constrained Route Distribution

Before you configure BGP: RT Constrained Route Distribution, you should understand how to configure the following:

- Multiprotocol Label Switching (MPLS) VPNs
- Route distinguishers (RDs)
- Route targets (RTs)
- Multiprotocol BGP (MBGP)

Restrictions for BGP: RT Constrained Route Distribution

BGP: RT Constrained Route Distribution constrains all VPN route advertisements.

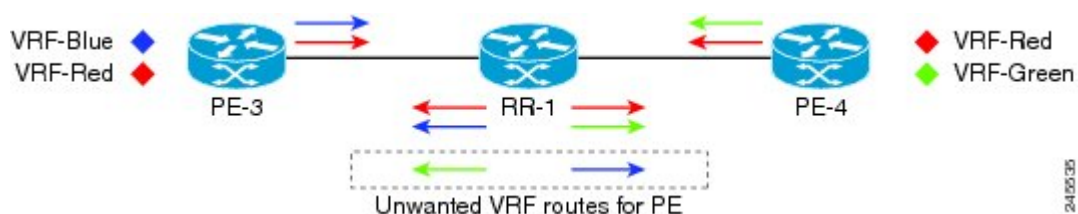
Information About BGP: RT Constrained Route Distribution

Problem That BGP: RT Constrained Route Distribution Solves

Some service providers have a large number of routing updates being sent from RRs to PEs, which can require extensive use of resources. A PE does not need routing updates for VRFs that are not on the PE; therefore, the PE determines that many routing updates it receives are "unwanted." The PE filters out the unwanted updates.

The figure below illustrates a scenario in which unwanted routing updates arrive at two PEs.

Figure 60: Unwanted Routing Updates at PE

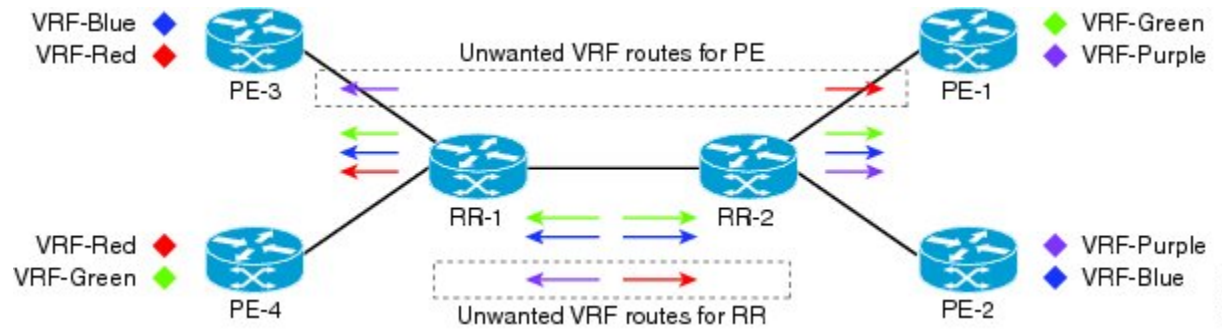


As shown in the figure above, a PE receives unwanted routes in the following manner:

- 1 PE-3 advertises VRF Blue and VRF Red routes to RR-1. PE-4 advertises VRF Red and VRF Green routes to RR-1.
- 2 RR-1 has all of the routes for all of the VRFs (Blue, Red, and Green).
- 3 During a route refresh or VRF provisioning, RR-1 advertises all of the VRF routes to both PE-3 and PE-4.
- 4 Routes for VRF Green are unwanted at PE-3. Routes for VRF Blue are unwanted at PE-4.

Now consider the scenario where there are two RRs with another set of PEs. There are unwanted routing updates from RRs to PEs and unwanted routing updates between RRs. The figure below illustrates a scenario in which unwanted routes arrive at an RR.

Figure 61: Unwanted Routing Updates at RR



As shown in the figure above, RR-1 and RR-2 receive unwanted routing updates in the following manner:

- 1 PE-3 and PE-4 advertise VRF Blue, VRF Red, and VRF Green VPN routes to RR-1.
- 2 RR-1 sends all of its VPN routes to RR-2.
- 3 VRF Red routes are unwanted on RR-2 because PE-1 and PE-2 do not have VRF Red.
- 4 Similarly, VRF Purple routes are unwanted on RR-1 because PE-3 and PE-4 do not have VRF Purple.

Hence, a large number of unwanted routes might be advertised among RRs and PEs. The BGP: RT Constrained Route Distribution feature addresses this problem by filtering unwanted routing updates.

Before the BGP: RT Constrained Route Distribution feature, the PE would filter the updates. With this feature, the burden is moved to the RR to filter the updates.

Benefits of BGP: RT Constrained Route Distribution

In MPLS L3VPNs, PE routers use BGP and route target (RT) extended communities to control the distribution of VPN routes to and from VRFs in order to separate the VPNs. PEs and Autonomous System Boundary Routers (ASBRs) commonly receive and then filter out the unwanted VPN routes.

However, receiving and filtering unwanted VPN routes is a waste of resources. The sender generates and transmits a VPN routing update and the receiver filters out the unwanted routes. Preventing the generation of VPN route updates would save resources.

Route Target Constrain (RTC) is a mechanism that prevents the propagation of VPN Network Layer Reachability Information (NLRI) from the RR to a PE that is not interested in the VPN. The feature provides considerable savings in CPU cycles and transient memory usage. RT constraint limits the number of VPN routes and describes VPN membership.

BGP RT-Constrain SAFI

The BGP: RT Constrained Route Distribution feature introduces the BGP RT-Constrain Subsequent Address Family Identifier (SAFI). The command to enter that address family is the **address-family rfilter unicast** command.

BGP: RT Constrained Route Distribution Operation

In order to filter out the unwanted routes described in the "Problem that BGP RT Constrained Route Distribution Solves" section on page 2, the PEs and RRs must be configured with the BGP: RT Constrained Route Distribution feature.

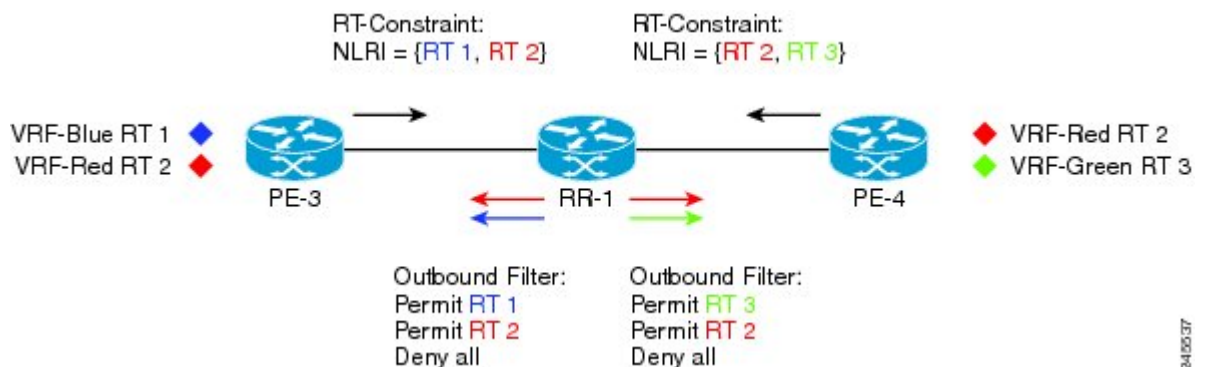
The feature allows the PE to propagate RT membership and use the RT membership to limit the VPN routing information maintained at the PE and RR. The PE uses an MP-BGP UPDATE message to propagate the membership information. The RR restricts advertisement of VPN routes based on the RT membership information it received.

This feature causes two exchanges to happen:

- The PE sends RT Constraint (RTC) Network Layer Reachability Information (NLRI) to the RR.
- The RR installs an outbound route filter.

The figure below illustrates the exchange of the RTC NLRI and the outbound route filter.

Figure 62: Exchange of RTC NLRI and Filter Between PE and RR



As shown in the figure above, the following exchange occurs between the PE and the RR:

- 1 PE-3 sends RTC NLRI (RT 1, RT 2) to RR-1.
- 2 PE-4 sends RTC NLRI (RT 2, RT 3) to RR-1.
- 3 RR-1 translates the NLRI into an outbound route filter and installs this filter (Permit RT 1, RT 2) for PE-3.
- 4 RR-1 translates the NLRI into an outbound route filter and installs this filter (Permit RT 2, RT 3) for PE-4.

RT Constraint NLRI Prefix

The format of the RT Constraint NLRI is a prefix that is always 12 bytes long, consisting of the following:

- 4-byte origin autonomous system
- 8-byte RT extended community value

The following are examples of RT Constraint prefixes:

- 65000:2:100:1
 - Origin autonomous system number is 65000
 - BGP Extended Community Type Code is 2
 - Route target is 100:1
- 65001:256:192.0.0.1:100
 - Origin ASN is 65001
 - BGP Extended Community Type Code is 256
 - Route target is 192.0.0.1:100
- 1.10:512:1.10:2
 - Origin ASN is 4-byte, unique 1.10
 - BGP Extended Community Type Code is 512
 - Route target is 1.10:2

To determine what the BGP Extended Community Type Code means, refer to RFC 4360, *BGP Extended Communities Attribute*. In the first example shown, a 2 translates in hexadecimal to 0x002. In RFC 4360, 0x002 indicates that the value that follows the type code will be a two-octet AS specific route target.

RT Constrained Route Distribution Process

This section shows the RT Constrained Route Distribution process. In this example has two CE routers in AS 100 that are connected to PE1. PE1 communicates with PE2, which is also connected to CE routers. Between the two PEs is a route reflector (RR). PE1 and PE2 belong to AS 65000.

The general process for the feature is as follows:

- 1 The user configures PE1 to activate its BGP peers under the **address-family rfilter unicast** command.
- 2 The user configures PE1 in AS 65000 with **route-target import 100:1**, for example.
- 3 PE1 translates that command to an RT prefix of 65000:2:100:1. The 65000 is the service provider's AS number; the 2 is the BGP Extended Communities Type Code; and the 100:1 is the CE's RT (AS number and another number).
- 4 PE1 advertises the RT Constrain (RTC) prefix of 65000:2:100:1 to its iBGP peer RR.

- 5 The RR installs RTC 65000:2:100:1 into the RTC RIB. Each VRF has its own RIB.
- 6 The RR also installs RTC 65000:2:100:1 into its outbound filter for the neighbor PE1.
- 7 A filter in the RR either permits or denies the RT. (The AS number is ignored because iBGP is operating in a single AS and does not need to track the AS number.)
- 8 The RR looks in its outbound filter and sees that it permits outbound VPN packets for RT 100:1 to PE1. So, the RR sends VPN update packet only with RT 100:1 to PE1 and denies VPN updates with any other RT.

Default RT Filter

The default RT filter has a value of zero and length of zero. The default RT filter is used:

- By a peer to indicate that the peer wants all of the VPN routes sent to it, regardless of the RT value.
- By the RR to request that the PE advertise all of its VPN routes to the RR.

The default RT filter is created by configuring the **neighbor default-originate** command under the **address-family rfilter unicast** command. On the RR it comes as default along with the configuration of route-reflector-client under the address-family rfilter.

How to Configure RT Constrained Route Distribution

Configuring Multiprotocol BGP on Provider Edge (PE) Routers and Route Reflectors

Perform this task to configure multiprotocol BGP (MP-BGP) connectivity on the PE routers and route reflectors.

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 vpnv4** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community** **extended**
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 100	Configures a BGP routing process and enters router configuration mode. • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the no form of the bgp default ipv4-unicast command if you are using this neighbor for MPLS routes only.
Step 5	neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor pp.0.0.1 remote-as 100	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group. • The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.
Step 6	address-family vpnv4 [unicast] Example: Device(config-router)# address-family vpnv4	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional unicast keyword specifies VPNv4 unicast address prefixes.
Step 7	neighbor {<i>ip-address</i> <i>peer-group-name</i>} send-community extended Example: Device(config-router-af)# neighbor pp.0.0.1 send-community extended	Specifies that a communities attribute should be sent to a BGP neighbor. • The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.

	Command or Action	Purpose
Step 8	neighbor <i>{ip-address peer-group-name}</i> activate Example: <pre>Device(config-router-af)# neighbor pp.0.0.1 activate</pre>	Enables the exchange of information with a neighboring BGP router. <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 9	end Example: <pre>Device(config-router-af)# end</pre>	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp ip-address events** command, where *ip-address* is the IP address of the neighbor.

Connecting the MPLS VPN Customers

To connect the MPLS VPN customers to the VPN, perform the following tasks:

Defining VRFs on PE Routers to Enable Customer Connectivity

To define virtual routing and forwarding (VRF) instances, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** *{import | export | both} route-target-ext-community*
6. **import map** *route-map*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip vrf vrf-name Example: Device(config)# ip vrf vpn1	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. <ul style="list-style-type: none"> • The <i>vrf-name</i> argument is the name assigned to a VRF.
Step 4	rd route-distinguisher Example: Device(config-vrf)# rd 100:1	Creates routing and forwarding tables. <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: <ul style="list-style-type: none"> • 16-bit AS number: your 32-bit number, for example, 101:3 • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1
Step 5	route-target {import export both} route-target-ext-community Example: Device(config-vrf)# route-target import 100:1	Creates a route-target extended community for a VRF. <ul style="list-style-type: none"> • The import keyword imports routing information from the target VPN extended community. • The export keyword exports routing information to the target VPN extended community. • The both keyword imports routing information from and exports routing information to the target VPN extended community. • The <i>route-target-ext-community</i> argument adds the RT extended community attributes to the VRF's list of import, export, or both (import and export) RT extended communities.
Step 6	import map route-map Example: Device(config-vrf)# import map vpn1-route-map	(Optional) Configures an import route map for a VRF. <ul style="list-style-type: none"> • The <i>route-map</i> argument specifies the route map to be used as an import route map for the VRF.

	Command or Action	Purpose
Step 7	exit Example: Device(config-vrf)# exit	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

To associate a VRF with an interface or subinterface on the PE routers, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface Ethernet 5/0	Specifies the interface to configure and enters interface configuration mode. <ul style="list-style-type: none"> • The <i>type</i> argument specifies the type of interface to be configured. • The <i>number</i> argument specifies the port, connector, or interface card number.
Step 4	ip vrf forwarding <i>vrf-name</i>	Associates a VRF with the specified interface or subinterface.

	Command or Action	Purpose
	Example: Device(config-if)# ip vrf forwarding vpn1	<ul style="list-style-type: none"> The <i>vrf-name</i> argument is the name assigned to a VRF.
Step 5	end Example: Device(config-if)# end	(Optional) Exits to privileged EXEC mode.

Configuring BGP as the Routing Protocol Between the PE and CE Routers

To configure PE-to-CE routing sessions using BGP, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i>	Configures a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config)# router bgp 100</pre>	<ul style="list-style-type: none"> The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	<p>address-family ipv4 [multicast unicast vrf vrf-name]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 vrf vpn1</pre>	<p>Specifies the IPv4 address family type and enters address family configuration mode.</p> <ul style="list-style-type: none"> The multicast keyword specifies IPv4 multicast address prefixes. The unicast keyword specifies IPv4 unicast address prefixes. The vrf vrf-name keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	<p>neighbor {ip-address peer-group-name} remote-as as-number</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor pp.0.0.1 remote-as 200</pre>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group. The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.
Step 6	<p>neighbor {ip-address peer-group-name} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor pp.0.0.1 activate</pre>	<p>Enables the exchange of information with a neighboring BGP router.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 7	<p>exit-address-family</p> <p>Example:</p> <pre>Device(config-router-af)# exit-address-family</pre>	<p>Exits address family configuration mode.</p>
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	<p>(Optional) Exits to privileged EXEC mode.</p>

Configuring RT Constraint on the PE

Perform this task on the PE to configure BGP: RT Constrained Route Distribution with the specified neighbor, and optionally verify that route target (RT) filtering is occurring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **address-family rtfiler unicast**
5. **neighbor {*ip-address* | *peer-group-name*} activate**
6. **neighbor {*ip-address* | *peer-group-name*} send-community extended**
7. **end**
8. **show ip bgp rtfiler all**
9. **show ip bgp rtfiler all summary**
10. **show ip bgp vpv4 all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 1	Configures a BGP routing process and enters router configuration mode.
Step 4	address-family rtfiler unicast Example: Device(config-router)# address-family rtfiler unicast	Specifies the RT filter address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> } activate Example: <pre>Device(config-router-af)# neighbor 10.0.0.1 activate</pre>	Enables the exchange of automated RT filter information with the specified BGP neighbor.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } send-community extended Example: <pre>Device(config-router-af)# neighbor pp.0.0.1 send-community extended</pre>	Specifies that a communities attribute should be sent to a BGP neighbor. <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 7	end Example: <pre>Device(config-router-af)# end</pre>	Exits configuration mode and returns to privileged EXEC mode.
Step 8	show ip bgp rtfiler all Example: <pre>Device# show ip bgp rtfiler all</pre>	(Optional) Displays all BGP RT filter information.
Step 9	show ip bgp rtfiler all summary Example: <pre>Device# show ip bgp rtfiler all summary</pre>	(Optional) Displays summary BGP RT filter information.
Step 10	show ip bgp vpnv4 all Example: <pre>Device# show ip bgp vpnv4 all</pre>	(Optional) Displays summary BGP VPNv4 information.

Configuring RT Constraint on the RR

Perform this task on the RR to configure BGP: RT Constrained Route Distribution with the specified neighbor, and optionally verify that route target (RT) filtering is occurring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family rtfiler unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client**
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
8. **end**
9. **show ip bgp rtfiler all**
10. **show ip bgp rtfiler all summary**
11. **show ip bgp vpnv4 all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 1	Configures a BGP routing process and enters router configuration mode.
Step 4	address-family rtfiler unicast Example: Device(config-router)# address-family rtfiler unicast	Specifies the RT filter address family type and enters address family configuration mode.
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 10.0.0.2 activate	Enables RT Constraint with the specified BGP neighbor.

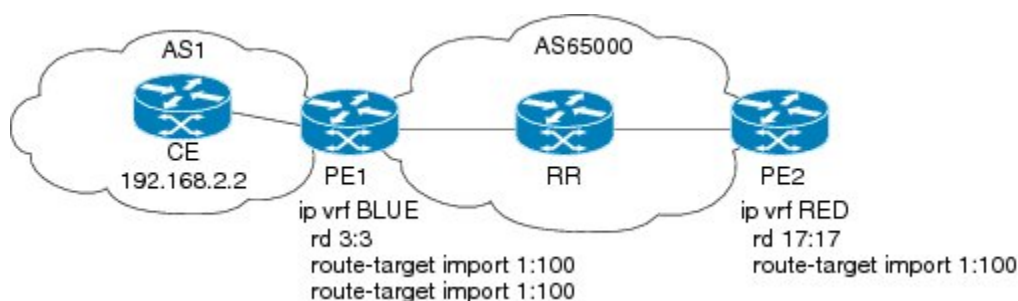
	Command or Action	Purpose
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-reflector-client</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.2 route-reflector-client</pre>	<p>Enables route-reflector-client functionality under RT Constraint with the specified BGP neighbor.</p> <ul style="list-style-type: none"> Note that the route-reflector-client under RT Constraint address-family comes with a default "neighbor 10.0.0.2 default-originate" functionality that automatically gets added to the BGP configuration. The reason to have this is to have the route-reflector get all the VPN prefixes from its peer.
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} send-community extended</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.2 send-community extended</pre>	<p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
Step 9	<p>show ip bgp rtfiler all</p> <p>Example:</p> <pre>Device# show ip bgp rtfiler all</pre>	<p>(Optional) Displays all BGP RT filter information.</p>
Step 10	<p>show ip bgp rtfiler all summary</p> <p>Example:</p> <pre>Device# show ip bgp rtfiler all summary</pre>	<p>(Optional) Displays summary BGP RT filter information.</p>
Step 11	<p>show ip bgp vpnv4 all</p> <p>Example:</p> <pre>Device# show ip bgp vpnv4 all</pre>	<p>(Optional) Displays summary BGP VPNv4 information.</p>

Configuration Examples for BGP: RT Constrained Route Distribution

Example: BGP RT Constrained Route Distribution Between a PE and RR

The following example provides the configurations of the routers in the figure below. PE1 and PE2 are each connected to the RR and belong to AS 65000.

Figure 63: BGP: RT Constrained Route Distribution Between a PE and RR



PE1 Configuration

```
ip vrf BLUE
 rd 3:3
  route-target export 1:100
  route-target import 1:100
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 65000
 neighbor 192.168.2.2 update-source Loopback0
 no auto-summary
!
 address-family vpv4
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
 exit-address-family
!
 address-family rtfiler unicast
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
 exit-address-family
!
 address-family ipv4 vrf BLUE
  redistribute static
 exit-address-family
!
ip route vrf BLUE 51.51.51.51 255.255.255.255 Null0
!
```

RR Configuration

```
!
router bgp 65000
 bgp log-neighbor-changes
```

```

neighbor 192.168.6.6 remote-as 65000
neighbor 192.168.6.6 update-source Loopback0
neighbor 192.168.7.7 remote-as 65000
neighbor 192.168.7.7 update-source Loopback0
!
address-family vpnv4
neighbor 192.168.6.6 activate
neighbor 192.168.6.6 send-community extended
neighbor 192.168.6.6 route-reflector-client
neighbor 192.168.7.7 activate
neighbor 192.168.7.7 send-community extended
neighbor 192.168.7.7 route-reflector-client
exit-address-family
!
address-family rtfiler unicast
neighbor 192.168.6.6 activate
neighbor 192.168.6.6 send-community extended
neighbor 192.168.6.6 route-reflector-client
neighbor 192.168.6.6 default-originate
neighbor 192.168.7.7 activate
neighbor 192.168.7.7 send-community extended
neighbor 192.168.7.7 route-reflector-client
neighbor 192.168.7.7 default-originate
exit-address-family
!

```

PE2 Configuration

```

!
ip vrf RED
rd 17:17
route-target export 150:15
route-target import 150:1
route-target import 1:100
!
router bgp 65000
bgp log-neighbor-changes
neighbor 192.168.2.2 remote-as 65000
neighbor 192.168.2.2 update-source Loopback0
neighbor 192.168.2.2 weight 333
no auto-summary
!
address-family vpnv4
neighbor 192.168.2.2 activate
neighbor 192.168.2.2 send-community extended
exit-address-family
!
address-family rtfiler unicast
neighbor 192.168.2.2 activate
neighbor 192.168.2.2 send-community extended
exit-address-family
!

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Related Topic	Document Title
L3VPNs and route targets	“Configuring MPLS Layer 3 VPNs” module in the <i>MPLS: Layer 3 VPNs Configuration Guide</i>
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4360	<i>BGP Extended Communities Attribute</i>
RFC 4684	<i>Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)</i>
RFC 5291	<i>Outbound Route Filtering Capability for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP: RT Constrained Route Distribution

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 57: Feature Information for BGP: RT Constrained Route Distribution

Feature Name	Releases	Feature Information
BGP: RT Constrained Route Distribution	15.1(1)S 15.2(3)T 15.2(4)S 15.1(1)SY	<p>BGP: Route Target (RT) Constrained Route Distribution is a feature that service providers can use in MPLS L3VPNs to reduce the number of unnecessary routes that RRs send to PEs, and thereby save resources.</p> <p>The following commands were introduced: address-family rtfiler unicast and show ip bgp rtfiler .</p> <p>In Cisco IOS Release 15.2(4)S, support was added for the Cisco 7200 series router.</p>



Configuring BGP Consistency Checker

The BGP Consistency Checker feature provides a way to identify certain types of BGP route inconsistencies with peers: next-hop label inconsistency, RIB-out inconsistency, and aggregation inconsistency. Upon finding such an inconsistency, the system sends a syslog error message and takes appropriate action if configured to do so.

- [Finding Feature Information, page 723](#)
- [Information About BGP Consistency Checker, page 723](#)
- [How to Configure BGP Consistency Checker, page 724](#)
- [Configuration Examples for BGP Consistency Checker, page 726](#)
- [Additional References, page 726](#)
- [Feature Information for BGP Consistency Checker, page 727](#)

Finding Feature Information

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

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

Information About BGP Consistency Checker

BGP Consistency Checker

A BGP route inconsistency with a peer occurs when an update or a withdraw is not sent to a peer, and black-hole routing can result. To identify that issue, BGP consistency checker was created as a low-priority process that does nexthop-label, RIB-out, and aggregation consistency checks at a configurable interval. When enabled,

BGP consistency checker is performed for all address families. Configuring BGP consistency checker is recommended.

Once the process identifies such an inconsistency, it will report the inconsistency with a syslog message and optionally take action if the **auto-repair** keyword is specified. The action taken depends on the type of inconsistency found.

- **Next-Hop Label Consistency Check**—When two paths have the same next hop because they are advertised by the same provider edge router (PE), they should also have the same next-hop label. If the labels are different, there is an inconsistency. If the **auto-repair** keyword is specified, the system will send a route-refresh request.
- **RIB-Out Consistency Check**—If a network passes an outbound policy and is not sent, or if a network does not pass an outbound policy and is sent, there is an inconsistency. If the **auto-repair** keyword is specified, the system will send a route-refresh request.
- **Aggregation Consistency Check**—If specific routes and the aggregated route become out of sync, an inconsistency can occur. Either the **error-message** keyword or the **auto-repair** keyword will trigger aggregation reevaluation.

In the unlikely event that you receive a syslog message about an inconsistency, notify your Cisco technical support representative with the syslog message exactly as it appears. The following are examples of such syslog messages:

- “Net 10.0.0.0/32 has Nexthop-Label inconsistency.”
- “Net 10.0.0.0/32 in IPv4 Unicast has rib-out inconsistency for update-group 4 - outbound-policy fails.”

How to Configure BGP Consistency Checker

Configure BGP Consistency Checker

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp consistency-checker** {**error-message** | **auto-repair**} [**interval** *minutes*]
5. **end**
6. **show ip bgp** [**vpn4** | **vpn6**] **all inconsistency nexthop-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 500</pre>	Configures a BGP routing process.
Step 4	<p>bgp consistency-checker {error-message auto-repair} [interval <i>minutes</i>]</p> <p>Example:</p> <pre>Router(config-router)# bgp consistency-checker auto-repair interval 720</pre>	<p>Enables BGP consistency checker.</p> <ul style="list-style-type: none"> The default interval is 1440 minutes (one day). The range is 5 to 1440 minutes.
Step 5	<p>end</p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Ends the current configuration and returns to privileged EXEC mode.
Step 6	<p>show ip bgp [vpnv4 vpnv6] all inconsistency nexthop-label</p> <p>Example:</p> <pre>Router# show ip bgp all inconsistency nexthop-label</pre>	<p>(Optional) Displays routes that have a nexthop-label inconsistency found.</p> <ul style="list-style-type: none"> This step is not part of configuring the feature; it is provided in case you receive a syslog message about a nexthop-label inconsistency and you want to display those routes.

Configuration Examples for BGP Consistency Checker

Example: Configuring BGP Consistency Checker

The following example configures BGP consistency checker with auto-repair at the default interval of one day:

```
router bgp 65000
  bgp consistency-checker auto-repair
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Consistency Checker

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 58: Feature Information for BGP Consistency Checker

Feature Name	Releases	Feature Information
BGP Consistency Checker	15.1(1)SY	<p>The BGP Consistency Checker feature provides a way to identify three types of BGP route inconsistencies with peers: next-hop label inconsistency, RIB-out inconsistency, and aggregation inconsistency. Upon finding such inconsistency, the system sends a syslog error message and takes appropriate action if configured to do so.</p> <p>The following command was introduced: bgp consistency-checker</p> <p>The following command was modified: show ip bgp vpv4.</p>



BGP MIB Support

The BGP MIB Support Enhancements feature introduces support in the CISCO-BGP4-MIB for new SNMP notifications.

- [Finding Feature Information, page 729](#)
- [Information About BGP MIB Support, page 729](#)
- [How to Enable BGP MIB Support, page 732](#)
- [Configuration Examples for BGP MIB Support, page 733](#)
- [Additional References, page 733](#)
- [Feature Information for BGP MIB Support, page 734](#)

Finding Feature Information

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

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

Information About BGP MIB Support

BGP MIB Support

The Management Information Base (MIB) that supports BGP is the CISCO-BGP4-MIB. The BGP MIB Support Enhancements feature introduces 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 be viewed in the output of the **`show ip bgp`** command.

BGP Prefix Threshold Notification Support

The `cbgpPrefixMaxThresholdExceed` and `cbgpPrfrefixMaxThresholdClear` 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 `cbgpPrfrefixMaxThresholdClear` 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 Enable BGP MIB Support

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

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p><code>snmp-server enable traps bgp</code> <code>[[state-changes [all] [backward-trans]</code> <code>[limited]] [threshold prefix]</code></p> <p>Example:</p> <pre>Device(config)# snmp-server enable traps bgp</pre>	<p>Enables BGP support for SNMP operations. Entering this command with no keywords or arguments enables support for all BGP events.</p> <ul style="list-style-type: none"> • The state-changes keyword is used to enable support for FSM transition events. • The all keyword enables support for FSM transitions events. • The backward-trans keyword enables support only for backward transition state change events. • The limited keyword enables support for backward transition state changes and established state events. • The threshold and prefix keywords are used to enable notifications when the configured maximum prefix limit is reached on the specified peer.

	Command or Action	Purpose
Step 4	exit Example: Device(config)# exit	Exits global configuration mode, and enters privileged EXEC mode.

Configuration Examples for BGP MIB Support

Example: Enabling BGP MIB Support

The following example enables SNMP support for all supported BGP events:

```
Device(config)# snmp-server enable traps bgp
```

The following verification example shows that SNMP support for BGP is enabled by displaying any lines in the running configuration file that include “snmp-server”:

```
Device# show run | include snmp-server
```

```
snmp-server enable traps bgp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	IP Routing: BGP Command Reference
MIB objects supported in CISCO-BGP-MIBv8.1	“Cisco-BGP-MIBv2” module in the <i>IP Routing: BGP Configuration Guide</i>
Information about SNMP and SNMP operations	SNMP Configuration Guide in the <i>Network Management Configuration Guide Library</i>

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP MIB Support

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 59: Feature Information for BGP MIB Support

Feature Name	Releases	Feature Information
BGP MIB Support Enhancements	12.0(26)S 12.2(25)S 12.3(7)T 12.2(33)SRA 12.2(22)SXH 15.0(1)SY	The BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following command was introduced: snmp-server enable traps bgp .



Cisco-BGP-MIBv2

The Cisco-BGP-MIBv2 feature supports CISCO-BGP-MIB, which allows MIBv2 traps to be generated. CISCO-BGP-MIB supports IPv4, IPv6, VPNv4, and VPNv6 address families. Its MIB objects and SNMP notifications can keep track of prefix counters and capabilities related to BGP sessions.

- [Finding Feature Information, page 735](#)
- [Information About Cisco-BGP-MIBv2, page 735](#)
- [How to Configure Cisco-BGP-MIBv2, page 740](#)
- [Configuration Examples for Cisco-BGP-MIBv2, page 741](#)
- [Additional References, page 741](#)
- [Feature Information for Cisco-BGP-MIBv2, page 742](#)

Finding Feature Information

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

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

Information About Cisco-BGP-MIBv2

Benefits of Cisco-BGP-MIBv2

The benefits of CISCO-BGP-MIB are as follows:

- MIB objects and SNMP notifications can keep track of prefix counters and capabilities related to BGP sessions.
- The following address families are supported: IPv4, IPv6, VPNv4, VPNv6.

- New traps are added to CISCO-BGP4-MIB that allow data to be returned for both IPv4 and IPv6 peers.

Cisco-BGP-MIBv2 Read-Only Objects

Cisco BGP Peer Table—cbgpPeer2Table

This table is an augmentation of the former Cisco BGP Peer Table, with flexible indexing to support both IPv4 and IPv6. This table contains one entry per BGP peer about the connections with BGP peers. MIB objects in this table are:

- cbgpPeer2Type
- cbgpPeer2RemoteAddr
- cbgpPeer2State
- cbgpPeer2AdminStatus
- cbgpPeer2NegotiatedVersion
- cbgpPeer2LocalAddr
- cbgpPeer2LocalPort
- cbgpPeer2LocalAs
- cbgpPeer2LocalIdentifier
- cbgpPeer2RemotePort
- cbgpPeer2RemoteAS
- cbgpPeer2RemoteIdentifier
- cbgpPeer2InUpdates
- cbgpPeer2OutUpdates
- cbgpPeer2InTotalMessages
- cbgpPeer2OutTotalMessages
- cbgpPeer2LastError
- cbgpPeer2FsmEstablishedTransitions
- cbgpPeer2FsmEstablishedTime
- cbgpPeer2ConnectRetryInterval
- cbgpPeer2HoldTime
- cbgpPeer2KeepAlive
- cbgpPeer2HoldTimeConfigured
- cbgpPeer2KeepAliveConfigured
- cbgpPeer2MinAsOriginationInterval
- cbgpPeer2MinRouteAdvertisementInterval
- cbgpPeer2InUpdateElapsedTime

- cbgpPeer2LastErrorTxt
- cbgpPeer2PrevState

BGP Session Capability Table—cbgpPeer2CapsTable

This table is an augmentation of the former Cisco BGP Session Capability Table, with flexible indexing to support both IPv4 and IPv6. BGP sessions can have multiple capabilities, which are sent to the peer in BGP OPEN messages. BGP_CAP_TYPE_AS4 is the only new capability support added from the previous version of cbgpPeerCapsTable. This table contains an entry corresponding to a capability code and an index. The MIB object in this table is:

- cbgpPeer2CapCode

Peer Address Family Table—cbgpPeer2AddrFamilyTable

This table is an augmentation of the former Cisco Peer Address Family Table, with flexible indexing to support both IPv4 and IPv6. This table contains supported address families. Only the following address families are supported: IPv4, VPNv4, IPv6, and VPNv6. Output is similar to the original MIB Object Identifier (OID): cbgpPeerAddrFamilyTable. The MIB object in this table is:

- cbgpPeer2AddrFamilyName

Prefix Counter Table—cbgpPeer2AddrFamilyPrefixTable

This table is an augmentation of the former Cisco Prefix Counter Table, with flexible indexing to support both IPv4 and IPv6. This table contains per peer route prefix counters for all of the supported address families. Only the following address families are supported: IPv4, VPNv4, IPv6, and VPNv6. Output is similar to the original MIB OID: cbgpPeerAddrFamilyPrefixTable. The MIB objects in this table are:

- cbgpPeer2AcceptedPrefixes
- cbgpPeer2DeniedPrefixes
- cbgpPeer2PrefixAdminLimit
- cbgpPeer2PrefixThreshold
- cbgpPeer2PrefixClearThreshold
- cbgpPeer2AdvertisedPrefixes
- cbgpPeer2SuppressedPrefixes
- cbgpPeer2WithdrawnPrefixes

cbgpNotifsEnable

This global object contains information about which traps are enabled. When the **snmp-server enable traps bgp cbgp2** command is configured, the following traps are sent:

cbgpPeer2FsmStateChange

This notification is an augmentation of the former FSM State Change Notification, with support for IPv6. This notification contains the following MIB objects:

- cbgpPeer2LastError
- cbgpPeer2State
- cbgpPeer2LastErrorTxt

- `cbgpPeer2PrevState`

cbgpPeer2BackwardTransition

This notification is an augmentation of the former Backward Transition Notification, with support for IPv6. This is sent when BGP FSM moves from a higher numbered state to a lower numbered state. This notification contains the following MIB objects:

- `cbgpPeer2LastError`
- `cbgpPeer2State`
- `cbgpPeer2LastErrorTxt`
- `cbgpPeer2PrevState`

cbgpPeer2PrefixThresholdExceeded

This notification is an augmentation of the former Prefix Threshold Exceeded Notification, with support for IPv6. This is sent when the prefix count for an address family on a BGP session exceeds the configured threshold value. This notification contains the following MIB objects:

- `cbgpPeer2PrefixAdminLimit`
- `cbgpPeer2PrefixThreshold`

cbgpPeer2PrefixThresholdClear

This notification is an augmentation of the former Prefix Threshold Clear Notification, with support for IPv6. It is sent when the prefix count drops below the clear trap limit for an address family on a BGP session after the generation of `cbgpPeer2PrefixThresholdExceeded` notification. This notification contains the following MIB objects:

- `cbgpPeer2PrefixAdminLimit`
- `cbgpPeer2PrefixClearThreshold`

cbgpPeer2BackwardTransNotification

This notification is generated when the BGP FSM moves from a higher numbered state to a lower numbered state. It contains the following MIB objects:

- `cbgpPeer2LastError`
- `cbgpPeer2State`
- `cbgpPeer2LastErrorTxt`
- `cbgpPeer2PrevState`

Cisco-BGP-MIBv2 Trap Objects

cbgpPeer2EstablishedNotification

This notification is generated when the BGP FSM enters the established state. The MIB objects are:

- `cbgpPeer2LastError`
- `cbgpPeer2State`

cbgpPeer2BackwardTransNotification

This notification is generated when the BGP FSM moves from a higher numbered state to a lower numbered state. The MIB objects are:

- cbgpPeer2LastError
- cbgpPeer2State

cbgpPeer2FsmStateChange

This notification is generated for every BGP FSM state change. The MIB objects are:

- cbgpPeer2LastError
- cbgpPeer2State
- cbgpPeer2LastErrorTxt
- cbgpPeer2PrevState

cbgpPeer2BackwardTransition

This notification is generated when the BGP FSM moves from a higher numbered state to a lower numbered state. The MIB objects are:

- cbgpPeer2LastError
- cbgpPeer2State
- cbgpPeer2LastErrorTxt
- cbgpPeer2PrevState

cbgpPeer2PrefixThresholdExceeded

This notification is generated when the prefix count exceeds the configured warning threshold on a session for an address family. The MIB objects are:

- cbgpPeer2PrefixAdminLimit
- cbgpPeer2PrefixThreshold

cbgpPeer2PrefixThresholdClear

This notification is generated when the prefix count drops below the configured clear threshold on a session for an address family once cbgpPeer2PrefixThresholdExceeded is generated. This notification is not generated if the peer session goes down after the generation of cbgpPeer2PrefixThresholdExceeded. The MIB objects are:

- cbgpPeer2PrefixAdminLimit
- cbgpPeer2PrefixClearThreshold

How to Configure Cisco-BGP-MIBv2

Enabling Cisco-BGP-MIBv2 Traps

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `snmp-server enable traps bgp cbgp2 [[state-changes [all] [backward-trans] [limited]] | [threshold prefix]]`
4. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp-server enable traps bgp cbgp2 [[state-changes [all] [backward-trans] [limited]] [threshold prefix]] Example: Device(config)# snmp-server enable traps bgp cbgp2	Enables generation of Cisco-BGP-MIBv2 traps. <ul style="list-style-type: none"> • The bgp keyword enables generation of original MIB traps. • The bgp cbgp2 keywords enable generation of MIBv2 traps. • The original MIB and MIBv2 can be enabled or disabled individually. If both the original MIB and MIBv2 are enabled, both traps are generated. If only one of the MIB versions is enabled, only that version of traps is generated.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Cisco-BGP-MIBv2

Example: Enabling Cisco-BGP-MIBv2

```
Device(config)# snmp-server enable traps bgp cbgp2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
CISCO-BGP4-MIB support	“BGP MIB Support” module in the <i>IP Routing: BGP Configuration Guide</i>
Information about SNMP and SNMP operations	SNMP Configuration Guide

MIBs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco-BGP-MIBv2

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 60: Feature Information for Cisco-BGP-MIBv2

Feature Name	Releases	Feature Information
Cisco-BGP-MIBv2	15.1(2)SY	<p>The Cisco-BGP-MIBv2 feature supports CISCO-BGP-MIB, which allows MIBv2 traps to be generated. CISCO-BGP-MIB supports IPv4, IPv6, VPNv4, and VPNv6 address families. Its MIB objects and SNMP notifications can keep track of prefix counters and capabilities related to BGP sessions.</p> <p>The following command was modified: snmp-server enable traps bgp.</p>



BGP Additional Paths

The BGP Additional Paths feature allows the advertisement of multiple paths through the same peering session for the same prefix without the new paths implicitly replacing any previous paths. This behavior promotes path diversity and reduces multi-exit discriminator (MED) oscillations.

- [Finding Feature Information, page 743](#)
- [Information About BGP Additional Paths, page 743](#)
- [How to Configure BGP Additional Paths, page 748](#)
- [Configuration Examples for BGP Additional Paths, page 759](#)
- [Additional References, page 761](#)
- [Feature Information for BGP Additional Paths, page 761](#)

Finding Feature Information

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

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

Information About BGP Additional Paths

Problem That Additional Paths Can Solve

BGP routers and route reflectors (RRs) propagate only their best path over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (this behavior is known as an implicit withdraw). The implicit withdraw can achieve better scaling, but at the cost of path diversity.

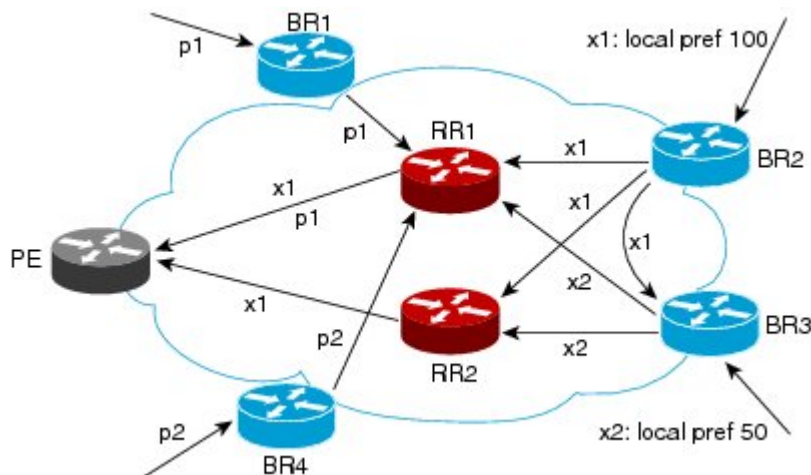
Path hiding can prevent efficient use of BGP multipath, prevent hitless planned maintenance, and can lead to MED oscillations and suboptimal hot-potato routing. Upon nexthop failures, path hiding also inhibits fast and local recovery because the network has to wait for BGP control plane convergence to restore traffic. The BGP Additional Paths feature provides a generic way of offering path diversity; the Best External or Best Internal features offer path diversity only in limited scenarios.

The BGP Additional Paths feature provides a way for multiple paths for the same prefix to be advertised without the new paths implicitly replacing the previous paths. Thus, path diversity is achieved instead of path hiding.

Path-Hiding Scenario

This section describes in more detail how path hiding can occur. In the following figure, we have prefix *p* with paths *p1* and *p2* advertised from BR1 and BR4 to RR1. RR1 selects the best path of the two and then advertises to PE only *p1*.

Figure 64: RR Hiding an Additional Path



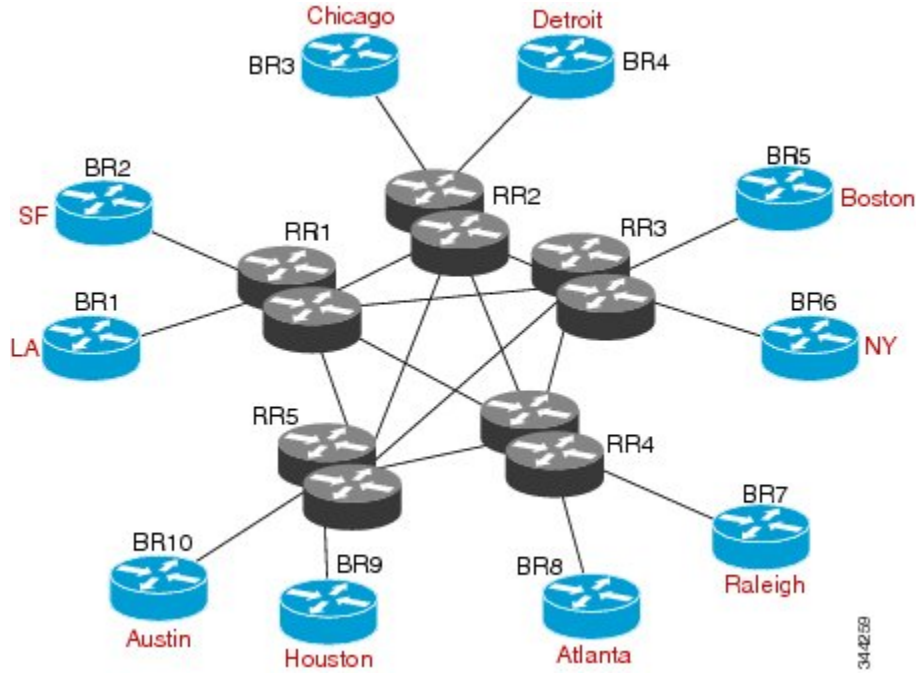
In the figure above, we also see prefix *x* with path *x1* being advertised from BR2 to BR3 (which has path *x2*) with local preference 100. BR3 also has path *x2*, but due to routing policy, BR3 will advertise to the RRs *x1* (not shown) instead of *x2*, and *x2* will be suppressed. A user could enable the advertisement of best external on BR3 and thereby advertise *x2* to the RRs, but, again, the RRs advertise only the best path.

Suboptimal Hot-Potato Routing Scenario

In order to minimize internal transport costs, transit ISPs try to forward packets to the closest exit point (according to Interior Gateway Protocol [IGP] cost). This behavior is known as hot-potato routing. In the distributed RR cluster model of the figure below, assume traffic coming from LA must go to Mexico. All links have the same IGP cost. If there are two exit points toward Mexico—one toward Austin and one toward Atlanta—the border router will try to send traffic to Austin based on the lower IGP cost from LA toward Austin than toward Atlanta. In a centralized RR model where the central RR resides where RR3 is (and RR1,

RR2, RR4, and RR5 do not exist), the closest exit point toward Mexico, as seen from RR3, might be Atlanta. Sending the traffic from LA toward Atlanta results in suboptimal hot-potato routing, which is not desirable.

Figure 65: Distributed RR Cluster

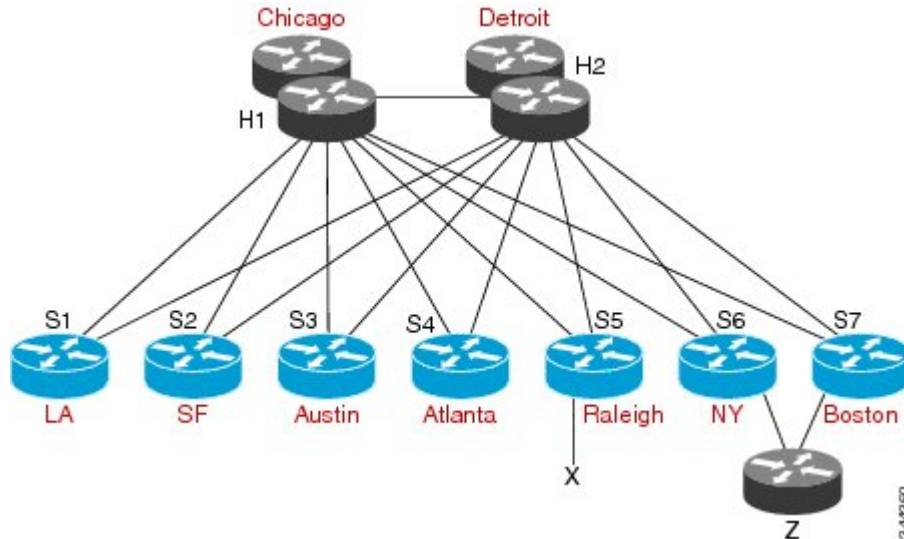


DMVPN Scenario

In Dynamic Multipoint Virtual Private Network (DMVPN) deployments, BGP is being used for scaling. In the figure below, Z is connected to both spokes S6 (NY) and S7 (Boston). The S7 links to the hubs have lower IGP costs than the S6 links to the hubs. There are physical links not shown that connect S5 to S6 and S6 to S7, with IGP costs lower than those to the hubs. Spokes S6 and S7 will send an update to both hubs H1 (Chicago) and H2 (Detroit). The RR hubs will then select the best path based on their lower IGP cost, which

might be S7. The spoke S5 (Raleigh) will receive two updates from the RRs for Z with S7 being the next hop, even though, in this scenario, it might be preferable to pick S6 (NY) as the next hop.

Figure 66: DMVPN Deployment



Benefits of BGP Additional Paths

BGP routers and route reflectors (RR) propagate only their best path over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (this is known as an implicit withdraw).

While this behavior may achieve better scaling, it can prevent path diversity, which tends to be poor or completely lost. The behavior in turn prevents efficient use of BGP multipath, prevents hitless planned maintenance, and can lead to multi-exit discriminator (MED) oscillations and suboptimal hot-potato routing. It also inhibits fast and local recovery upon nexthop failures, because the network has to wait for BGP control plane convergence to restore traffic.

The BGP Additional Paths feature is a BGP extension that allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous paths. This behavior promotes path diversity and reduces MED oscillations.

BGP Additional Paths Functionality

The BGP Additional Paths feature is implemented by adding a path identifier to each path in the NLRI. The path identifier (ID) can be considered as something similar to a route distinguisher (RD) in VPNs, except that a path ID can apply to any address family. Path IDs are unique to a peering session and are generated for each network. The path identifier is used to prevent a route announcement from implicitly withdrawing the previous one. The Additional Paths feature allows the advertisement of more paths, in addition to the bestpath. The Additional Paths feature allows the advertisement of multiple paths for the same prefix, without the new paths implicitly replacing any previous paths.

The BGP Additional Paths feature requires the user to take three general steps:

1. Specify whether the device can send, receive, or send and receive additional paths. This is done at the address family level or the neighbor level, and is controlled by either the **bgp additional-paths {send [receive] | receive}** command or the **neighbor additional-paths {send [receive] | receive}** command, respectively. During session establishment, two BGP neighbors negotiate the Additional Path capabilities (whether they can send and/or receive) between them.
2. Select a set or sets of candidate paths for advertisement by specifying selection criteria (using the **bgp additional-paths select** command).
3. Advertise for a neighbor a set or sets of additional paths from the candidate paths marked (using the **neighbor advertise additional-paths** command).

To send or receive additional paths, the Additional Path capability must be negotiated. If it isn't negotiated, even if the selection criteria are such that more than the bestpath is marked and the neighbor is configured to advertise the marked paths, the selections would be useless because without the capability negotiated, only the bestpath can be sent.

Configuring BGP to send or receive additional paths triggers negotiation of additional path capability with the device's peers. Neighbors that have negotiated the capability will be grouped together in an update group (if other update group policies allow), and in a separate update group from those peers that have not negotiated the capability. Therefore, additional path capability causes the neighbor's update group membership to be recalculated.

Additional Path Selection

There are three path selection (path marking) policies, and they are not mutually exclusive. They are specified per address family, using the **bgp additional-paths select** command. They are:

- **best 2** or **best 3** (**best 2** means the bestpath and 2nd best path; the 2nd best path is the one computed by eliminating best-path from the best-computation algorithm. Similarly, **best 3** means the bestpath, 2nd best path, and 3rd best path; the 3rd best path is the one computed by eliminating bestpath and 2nd best path from the best-computation algorithm.)
- **group-best** (calculates the group-best for prefixes during bestpath calculation; described further below)
- **all** (all paths with unique next hops are eligible for selection)

Definition of the group-best Selection

The **group-best** keyword is part of the following commands:

- **advertise additional-paths**
- **bgp additional-paths select**
- **match additional-paths advertise-set**
- **neighbor advertise additional-paths**

The **group-best** is the set of paths that are the best paths from the paths of the same AS. For example, suppose there are three autonomous systems: AS 100, 200, and 300. Paths p101, p102, and p103 are from AS 100; p201, p202, and p203 are from AS200; and p301, p302, and p303 are from AS300. If we run the BGP bestpath algorithm on the paths from each AS, the algorithm will select one bestpath from each set of paths from that AS. Assuming p101 is the best from AS100, p201 is the best from AS200, and p301 is the best from AS300, then the **group-best** is the set of p101, p201, and p301.

Advertise a Subset of the Paths Selected

Take care when you select a set of paths but want to advertise a different set of paths. If the set of paths you want to advertise is not a subset of the selected paths, then you will not advertise the paths you want advertised.

The following example configures the additional paths selected to be the group-best and all selections. However, the paths configured to be advertised to the neighbor are the best 3 paths. Because the selection and advertise policy are not the same, the subsequent message is displayed. In these cases, only the bestpath is advertised.

```
Device(config)# router bgp 100
Device(config-router)# address-family ipv4
Device(config-router-af)# bgp additional-paths send receive
Device(config-router-af)# bgp additional-paths select group-best all
Device(config-router-af)# neighbor 192.168.2.2 advertise additional-paths best 3
% BGP: AF level 'bgp additional-paths select' more restrictive than advertising policy.
This is a reminder that AF level additional-path select commands are needed.
```

How to Configure BGP Additional Paths

Configuring Additional Paths per Address Family

To select which paths are candidates to be additional paths, you can perform any combination of Steps 6, 7, and 8, as long as you perform at least one of those steps.

If you want to disable additional paths per neighbor, see the “Disabling Additional Paths per Neighbor” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **address-family ipv4 [unicast | multicast]**
5. **bgp additional-paths {send [receive] | receive}**
6. **bgp additional-paths select group-best**
7. **bgp additional-paths select best *number***
8. **bgp additional-paths select all**
9. **neighbor {*ip-address* | *ipv6-address* | *peer-group-name* } advertise additional-paths [best *number*] [group-best] [all]**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	address-family ipv4 [unicast multicast] Example: Device(config-router)# address-family ipv4	Enters address family configuration mode. <ul style="list-style-type: none"> The following address families are supported: IPv4 unicast, IPv4 multicast, IPv4 unicast + label, IPv6 unicast, IPv6 multicast, and IPv6 multicast + label.
Step 5	bgp additional-paths {send [receive] receive} Example: Device(config-router-af)# bgp additional-paths send receive	Enables BGP additional paths to be sent only, received only, or sent and received, after negotiation with the neighbor is completed. <ul style="list-style-type: none"> This example enables additional paths to be sent and received.
Step 6	bgp additional-paths select group-best Example: Device(config-router-af)# bgp additional-paths select group-best	(Optional) Calculates the group-best for prefixes during bestpath calculation.
Step 7	bgp additional-paths select best <i>number</i> Example: Device(config-router-af)# bgp additional-paths select best 3	(Optional) Calculates the specified number of best paths, including the advertisement of the bestpath. <ul style="list-style-type: none"> The value of <i>number</i> can be 2 or 3.
Step 8	bgp additional-paths select all Example: Device(config-router-af)# bgp additional-paths select all	(Optional) Specifies that all paths with unique next hops are eligible for selection.
Step 9	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } advertise additional-paths [best <i>number</i>] [group-best] [all]	Specifies which selection methods control the additional paths that are advertised to the neighbor.

	Command or Action	Purpose
	Example: <pre>Device(config-router-af)# neighbor 192.168.0.1 advertise additional-paths best 3 group-best all</pre>	
Step 10	end Example: <pre>Device(config-router-af)# end</pre>	(Optional) Exits to privileged EXEC mode.

Configuring Additional Paths per Neighbor

To select which paths are candidates to be additional paths, you can perform any combination of Steps 6, 7, and 8, as long as you perform at least one of those steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**unicast** | **multicast**]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **additional-paths** {**send** [**receive**] | **receive**}
6. **bgp additional-paths select group-best**
7. **bgp additional-paths select best** *number*
8. **bgp additional-paths select all**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **advertise additional-paths** [**best** *number*] [**group-best**] [**all**]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 65000</pre>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	<p>address-family ipv4 [unicast multicast]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Enters address family configuration mode.</p> <ul style="list-style-type: none"> The following address families are supported: IPv4 unicast, IPv4 multicast, IPv4 unicast + label, IPv6 unicast, IPv6 multicast, and IPv6 multicast + label.
Step 5	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} additional-paths {send [receive] receive}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.168.1.2 additional-paths send receive</pre>	<p>Enables the neighbor to send or receive additional paths after negotiation is completed.</p> <ul style="list-style-type: none"> This example enables the neighbor to send and receive additional paths. Note that this command overrides any send or receive capability that might have been configured at the address-family level.
Step 6	<p>bgp additional-paths select group-best</p> <p>Example:</p> <pre>Device(config-router-af)# bgp additional-paths select group-best</pre>	(Optional) Calculates the group-best for prefixes during bestpath calculation.
Step 7	<p>bgp additional-paths select best <i>number</i></p> <p>Example:</p> <pre>Device(config-router-af)# bgp additional-paths select best 3</pre>	<p>(Optional) Calculates the specified number of best paths, including the selection of the bestpath.</p> <ul style="list-style-type: none"> The value of <i>number</i> can be 2 or 3.
Step 8	<p>bgp additional-paths select all</p> <p>Example:</p> <pre>Device(config-router-af)# bgp additional-paths select all</pre>	(Optional) Specifies that all paths with unique next hops are eligible for selection.

	Command or Action	Purpose
Step 9	neighbor <i>{ip-address ipv6-address peer-group-name}</i> advertise additional-paths [best number] [group-best] [all] Example: Device(config-router-af)# neighbor 192.168.1.2 advertise additional-paths best 3 group-best all	Specifies the selection methods that control which additional paths are advertised for the neighbor.
Step 10	end Example: Device(config-router-af)# end	(Optional) Exits to privileged EXEC mode.

Configuring Additional Paths Using a Peer Policy Template

In this configuration task example, the capability to send and receive additional paths and the selection criteria are configured for the address family, and then the template is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 unicast**
5. **bgp additional-paths** {send [receive] | receive}
6. **bgp additional-paths select** [**best number**] [**group-best**] [**all**]
7. **template peer-policy** *policy-template-name*
8. **additional-paths** {send [receive] | receive}
9. **advertise additional-paths** [**best number**] [**group-best**] [**all**]
10. **exit**
11. **address-family ipv4 unicast**
12. **neighbor** *{ip-address | ipv6-address | peer-group-name}* **remote-as** *autonomous-system-number*
13. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	address-family ipv4 unicast Example: Device(config-router)# address-family ipv4 unicast	Configures the IPv4 address family.
Step 5	bgp additional-paths {send [receive] receive} Example: Device(config-router)# bgp additional-paths send receive	Enables BGP additional paths to be sent only, received only, or sent and received for the peers in the address family.
Step 6	bgp additional-paths select [best <i>number</i>] [group-best] [all] Example: Device(config-router)# bgp additional-paths select best 3 group-best all	Causes the system to calculate BGP additional paths that can be candidates for advertisement in addition to a bestpath.
Step 7	template peer-policy <i>policy-template-name</i> Example: Device(config-router)# template peer-policy rr-client-pt1	Enters policy-template configuration mode and creates a peer policy template.
Step 8	additional-paths {send [receive] receive} Example: Device(config-router-ptmp)# additional-paths send receive	Enables BGP additional paths to be sent only, received only, or sent and received for the peers covered by the peer policy template.

	Command or Action	Purpose
Step 9	advertise additional-paths [<i>best number</i>] [<i>group-best</i>] [all] Example: <pre>Device(config-router-ptmp)# advertise additional-paths best 3 group-best all</pre>	Specifies the selection methods that control which additional paths are advertised for the peers covered by the peer policy template.
Step 10	exit Example: <pre>Device(config-router-ptmp)# exit</pre>	Exits policy-template configuration mode and returns to router configuration mode.
Step 11	address-family ipv4 unicast Example: <pre>Device(config-router)# address-family ipv4 unicast</pre>	Configures the IPv4 address family.
Step 12	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: <pre>Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000</pre>	Adds an entry to the BGP neighbor table.
Step 13	neighbor <i>ip-address</i> inherit peer-policy <i>policy-template-name</i> Example: <pre>Device(config-router-af)# neighbor 192.168.1.1 inherit peer-policy rr-client-pt1</pre>	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.
Step 14	end Example: <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Filtering and Setting Actions for Additional Paths

You can optionally use a route map to filter the paths to be advertised by matching on the tags of additional paths that are candidates to be advertised. (These tags are the advertise-sets that are configured with the **bgp additional-paths select** command.) Paths that have the same path marking (tag) as the marking that is

configured in the **match additional-paths advertise-set** command match the route map entry (and are permitted or denied).

You can also optionally set one or more actions to take for those paths that pass the route map. This task happens to use the **set metric** command to illustrate using a route map with the **match additional-paths advertise-set** command. Of course, other **set** commands are available that are not shown in this task.

Why set a metric for paths marked with **all** (all paths with a unique next hop)? Suppose the neighbor 2001:DB8::1037 is receiving the same route from different neighbors. Routes received from the local device have a metric of 565 and routes from another device perhaps have a metric of 700. Routes with metric 565 will have precedence over the routes with metric 700.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match additional-paths advertise-set** [**best number**] [**best-range** *start-range end-range*] [**group-best**] [**all**]
5. **set metric** *metric-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# route-map additional_path1 permit 10	Creates a route map.
Step 4	match additional-paths advertise-set [best number] [best-range <i>start-range end-range</i>] [group-best] [all] Example: Device(config-route-map)# match additional-paths advertise-set best 3	Matches on any path that is tagged with the specified path selection policy. <ul style="list-style-type: none"> • You must specify at least one selection method; you can specify more than one selection method in the command. • Specifying best number is incompatible with specifying best-range.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Specifying best 1 will match only the bestpath. • Specifying best-range 1 1 will match only the bestpath. • Only one match additional-paths advertise-set command is allowed per route map. A subsequent match additional-paths advertise-set command will overwrite the previous command.
Step 5	set metric <i>metric-value</i> Example: Device(config-route-map)# set metric 500	Sets the metric of the additional paths that pass the match criteria. <ul style="list-style-type: none"> • Note that other set commands can be used to take action on the paths that pass the route map. This example happens to use the set metric command.

What to Do Next

After creating the route map, you would reference the route map in the **neighbor route-map out** command. Thus, the route map is applied to paths being advertised (outgoing) to neighbors. Then you would use the **neighbor advertise additional-paths** command to advertise the additional paths. See the “Example: BGP Additional Paths” section to see the route map in context.

Displaying Additional Path Information

Perform either Step 2 or Step 3 in this task to see information about BGP additional paths.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address*]
3. **show ip bgp** [*network*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	show ip bgp neighbors [<i>ip-address</i>] Example: Device# show ip bgp neighbors 192.168.1.1	Displays the capabilities of the neighbor to send and receive additional paths.
Step 3	show ip bgp [<i>network</i>] Example: Device# show ip bgp 192.168.0.0	Displays the additional path selections and path ID for the network.

Disabling Additional Paths per Neighbor

If you had configured the sending or receiving of additional paths on a per neighbor basis (with the **neighbor additional-paths** command), and you wanted to disable that functionality, you would use the **no neighbor additional-paths** command.

However, if you had configured the sending or receiving of additional paths for an address family (with the **bgp additional-paths** command), and you wanted to disable that functionality for a neighbor, you would use the **neighbor additional-paths disable** command. Disabling additional paths also works if the functionality was inherited from a template.

Perform this task to disable additional path capability for a neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*unicast* | *multicast*]
5. **bgp additional-paths** {*send* [*receive*] | *receive*}
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **additional-paths disable**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	address-family ipv6 [unicast multicast] Example: Device(config-router)# address-family ipv6 unicast	Enters address family configuration mode.
Step 5	bgp additional-paths {send [receive] receive} Example: Device(config-router-af)# bgp additional-paths send receive	Enables BGP additional paths to be sent or received for the neighbors in the address family.
Step 6	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} additional-paths disable Example: Device(config-router-af)# neighbor 2001:DB8::1 additional-paths disable	Disables BGP additional paths from being sent to or received from the specified neighbor. <ul style="list-style-type: none"> • The additional path functionality is still enabled for the rest of the neighbors in the address family.
Step 7	end Example: Device(config-router-af)# end	(Optional) Exits to privileged EXEC mode.

Configuration Examples for BGP Additional Paths

Example: BGP Additional Path Send and Receive Capabilities

In this example, R1's address is 192.168.1.1; its neighbor is R2, which has address 192.168.1.2. Updates are sent from R2 to R1 with additional-paths (all paths advertised). Updates are sent from R1 to R2 with only the classic BGP bestpath advertised because R2 is only able to send additional paths, not receive additional paths.

R1

```
router bgp 1
 address-family ipv4 unicast
  bgp additional-paths select all
  neighbor 192.168.1.2 additional-paths send receive
  neighbor 192.168.1.2 advertise additional-paths all
```

R2

```
router bgp 2
 address-family ipv4 unicast
  bgp additional-paths select all
  neighbor 192.168.1.1 additional-paths send
  neighbor 192.168.1.1 advertise additional-paths all
```

Example: BGP Additional Paths

In the following example, for every address family, there are one or more eBGP neighbors not shown in the configuration that are sending routes to the local device. The eBGP routes learned from those neighbors are advertised toward the neighbors shown in the configuration below and the path attributes are changed. The example configures that:

- The route map called `add_path1` specifies that all the paths are advertised toward neighbor 192.168.101.15, but any path that is marked with **best 2** will have its metric set to 780 before being sent toward that neighbor.
- The route map called `add_path2` specifies that any path that is marked with **best 3** will have its metric set to 640 and will be advertised toward neighbor 192.168.25.
- The route map called `add_path3` specifies that any path that is marked with **group-best** will have its metric set to 825 and will be advertised toward neighbor 2001:DB8::1045.
- In the IPv6 multicast address family, all paths are candidates to be advertised and will be advertised toward neighbor 2001:DB8::1037.

```
router bgp 1
 neighbor 192.168.101.15 remote-as 1
 neighbor 192.168.101.25 remote-as 1
 neighbor 2001:DB8::1045 remote-as 1
 neighbor 2001:DB8::1037 remote-as 1
 !
 address-family ipv4 unicast
  bgp additional-paths send receive
  bgp additional-paths select all best 3 group-best
  neighbor 192.168.101.15 activate
  neighbor 192.168.101.15 route-map add_path1 out
  neighbor 192.168.101.15 advertise additional-paths best 2
 exit-address-family
 !
 address-family ipv4 multicast
```

Example: Neighbor Capabilities Override Address Family Capabilities

```

    bgp additional-paths send receive
    bgp additional-paths select all best 3 group-best
    neighbor 192.168.101.25 activate
    neighbor 192.168.101.25 route-map add_path2 out
    neighbor 192.168.101.25 advertise additional-paths best 3
  exit-address-family
!
  address-family ipv6 unicast
    bgp additional-paths send receive
    bgp additional-paths select group-best
    neighbor 2001:DB8::1045 activate
    neighbor 2001:DB8::1045 route-map add_path3 out
    neighbor 2001:DB8::1045 advertise additional-paths all group-best
  exit-address-family
!
  address-family ipv6 multicast
    bgp additional-paths send receive
    bgp additional-paths select all
    neighbor 2001:DB8::1037 activate
    neighbor 2001:DB8::1037 route-map add_path4 out
    neighbor 2001:DB8::1037 advertise additional-paths all
  exit-address-family
!
  route-map add_path1 permit 10
  match additional-paths advertise-set best 2
  set metric 780
  route-map add_path1 permit 20
!
  route-map add_path2 permit 10
  match additional-paths advertise-set best 3
  set metric 640
!
  route-map add_path3 permit 10
  match additional-paths advertise-set group-best
  set metric 825
!

```

Example: Neighbor Capabilities Override Address Family Capabilities

In the following example, the receive-only capability of the neighbor overrides the send and receive capability of the address family:

```

router bgp 65000
  address-family ipv6 multicast
    bgp additional-paths send receive
    bgp additional-paths select group-best
    neighbor 2001:DB8::1037 activate
    neighbor 2001:DB8::1037 additional-paths receive
    neighbor 2001:DB8::1037 advertise additional-paths group-best
  !

```

Example: BGP Additional Paths Using a Peer Policy Template

```

router bgp 45000
  address-family ipv4 unicast
    bgp additional-paths send receive
    bgp additional-paths select all group-best best 3
    template peer-policy rr-client-pt1
    additional-paths send receive
    advertise additional-paths group-best best 3
  exit
  address-family ipv4 unicast
    neighbor 192.168.1.1 remote-as 45000
    neighbor 192.168.1.1 inherit peer-policy rr-client-pt1

```

end

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 3107	<i>Carrying Label Information in BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol (BGP-4)</i>
RFC 4760	<i>Multiprotocol Extensions for BGP-4</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Additional 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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 61: Feature Information for BGP Additional Paths

Feature Name	Releases	Feature Information
BGP Additional Paths	15.2(1)SY	<p>The BGP Additional Paths feature allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous paths.</p> <p>The following commands were introduced:</p> <ul style="list-style-type: none"> • additional-paths • advertise additional-paths • bgp additional-paths • bgp additional-paths select • match additional-paths advertise-set • neighbor additional-paths • neighbor advertise additional-paths <p>The following commands were modified:</p> <ul style="list-style-type: none"> • show ip bgp • show ip bgp neighbors



BGP Attribute Filter and Enhanced Attribute Error Handling

The BGP Attribute Filter feature allows you to “treat-as-withdraw” updates that contain specific path attributes. The prefixes contained in the update are removed from the routing table. The feature also allows you to remove specific path attributes from incoming updates. Both behaviors provide an increased measure of security. The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to errors from any malformed update, thereby saving resources.

- [Finding Feature Information, page 763](#)
- [Information About BGP Attribute Filtering, page 763](#)
- [How to Filter BGP Path Attributes, page 765](#)
- [Configuration Examples for BGP Attribute Filter, page 768](#)
- [Additional References, page 769](#)
- [Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling, page 770](#)

Finding Feature Information

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

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

Information About BGP Attribute Filtering

BGP Attribute Filter and Enhanced Attribute Error Handling

The BGP Attribute Filter feature provides two ways to achieve an increased measure of security:

- The feature allows you to treat-as-withdraw an Update coming from a specified neighbor if the Update contains a specified attribute type. When an Update is treat-as-withdraw, the prefixes in the Update are removed from the BGP routing table (if they existed in the routing table).
- The feature also allows you to drop specified path attributes from an Update, and then the system processes the rest of the Update as usual.

The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to a malformed Update. The malformed Update is treat-as-withdraw and does not cause the BGP session to be reset. This feature is enabled by default, but can be disabled.

The features are implemented in the following order:

- 1 Received Updates that contain user-specified path attributes are treat-as-withdraw (as long as the NLRI can be parsed successfully). If there is an existing prefix in the BGP routing table, it will be removed. The **neighbor path-attribute treat-as-withdraw** command configures this feature.
- 2 User-specified path attributes are discarded from received Updates, and the rest of the Update is processed normally. The **neighbor path-attribute discard** command configures this feature.
- 3 Received Updates that are malformed are treat-as-withdraw. This feature is enabled by default; it can be disabled by configuring the **no bgp enhanced-error** command.

Details About Specifying Attributes as Treat-as-Withdraw

Attribute types 1, 2, 3, 4, 8, 14, 15, and 16 cannot be configured for path attribute treat-as-withdraw.

Attribute type 5 (localpref), type 9 (Originator), and type 10 (Cluster-id) can be configured for treat-as-withdraw for eBGP neighbors only.

Configuring path attributes to be treated as withdrawn will trigger an inbound Route Refresh to ensure that the routing table is up to date.

Details About Specifying Attributes as Discard

Attribute types 1, 2, 3, 4, 8, 14, 15, and 16 cannot be configured for path attribute discard.

Attribute type 5 (localpref), type 9 (Originator), and type 10 (Cluster-id) can be configured for discard for eBGP neighbors only.

Configuring path attributes to be discarded will trigger an inbound Route Refresh to ensure that the routing table is up to date.

Details About Enhanced Attribute Error Handling

If a malformed Update is received, it is treat-as-withdraw to prevent peer sessions from flapping due to the processing of BGP path attributes. This feature applies to eBGP and iBGP peers. This feature is enabled by default; it can be disabled.

If the BGP Enhanced Attribute Error Handling feature is enabled or disabled, BGP places the MP_REACH attribute (attribute 14) at the beginning of an attribute list while formatting an update. Enhanced attribute error handling functions more easily when the MP_REACH attribute is at the beginning of the attribute list.

How to Filter BGP Path Attributes

Treat-as-Withdraw BGP Updates Containing a Specified Path Attribute



Note Performing this task will trigger an inbound Route Refresh to ensure that the routing table is up to date.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **path-attribute treat-as-withdraw** {*attribute-value* | **range** *start-value end-value*} **in**
5. Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> } path-attribute treat-as-withdraw { <i>attribute-value</i> range <i>start-value end-value</i> } in Example: Device(config-router)# neighbor 2001:DB8:1::1 path-attribute treat-as-withdraw 100 in	Treat-as-withdraw any incoming Update messages that contain the specified path attribute or range of path attributes. • Any prefixes in an Update that is treat-as-withdraw are removed from the BGP routing table.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The specific attribute value and the range of attribute values are independent of each other.
Step 5	Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.	
Step 6	end Example: Device(config-router)# end	Exits to privileged EXEC mode.

Discarding Specific Path Attributes from an Update Message



Note Performing this task will trigger an inbound Route Refresh to ensure that the routing table is up to date.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **path-attribute discard** {*attribute-value* | **range** *start-value end-value*} **in**
5. Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 6500</pre>	Configures a BGP routing process and enters router configuration mode.
Step 4	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i>} path-attribute discard {<i>attribute-value</i> range <i>start-value end-value</i>} in</p> <p>Example:</p> <pre>Device(config-router)# neighbor 2001:DB8:1::1 path-attribute discard 128 in</pre>	Drops specified path attributes from Update messages from the specified neighbor.
Step 5	<p>Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.</p> <p>Example:</p>	
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits to privileged EXEC mode.

Displaying Withdrawn or Discarded Path Attributes

Perform any of these steps in any order to display information about treat-as-withdraw, discarded, or unknown path attributes. You can use the **show ip bgp** command with any address family that BGP supports, such as **show ip bgp ipv4 multicast**, **show ip bgp ipv6 unicast**, etc.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbor** [*ip-address* | *ipv6-address*]
3. **show ip bgp path-attribute unknown**
4. **show ip bgp path-attribute discard**
5. **show ip bgp vpv4 all** *prefix*
6. **show ip bgp neighbors** *prefix*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show ip bgp neighbor [<i>ip-address</i> <i>ipv6-address</i>] Example: Device# show ip bgp neighbor 2001:DB8:1::1	(Optional) Displays the configured discard and treat-as-withdraw attribute values for the neighbor, counts of Updates with such attributes discarded or treat-as-withdraw, and the count of malformed treat-as-withdraw Updates.
Step 3	show ip bgp path-attribute unknown Example: Device# show ip bgp path-attribute unknown	(Optional) Displays all prefixes that have an unknown attribute.
Step 4	show ip bgp path-attribute discard Example: Device# show ip bgp path-attribute discard	(Optional) Displays all prefixes for which an attribute has been discarded.
Step 5	show ip bgp vpnv4 all <i>prefix</i> Example: Device# show ip bgp vpnv4 all 192.168.1.0	(Optional) Displays the unknown attributes and discarded attributes associated with a prefix.
Step 6	show ip bgp neighbors <i>prefix</i> Example: Device# show ip bgp neighbors 192.168.1.0	(Optional) Displays the configured discard and treat-as-withdraw attributes associated with a prefix.

Configuration Examples for BGP Attribute Filter

Examples: Withdraw Updates Based on Path Attribute

The following example shows how to configure the device to treat-as-withdraw any Update messages from the specified neighbor that contain the unwanted path attribute 100 or 128:

```
router bgp 65600
 neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 100 in
 neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 128 in
```

The following example shows how to configure the device to treat-as-withdraw any Update messages from the specified neighbor that contain the unwanted path attributes in the range from 21 to 255:

```
router bgp 65600
 neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 21 255 in
```

Examples: Discard Path Attributes from Updates

The following example shows how to configure the device to discard path attributes 100 and 128 from incoming Update messages from the specified neighbor. The rest of the Update message will be processed as usual.

```
router bgp 65600
 neighbor 2001:DB8:1::1 path-attribute discard 100 in
 neighbor 2001:DB8:1::1 path-attribute discard 128 in
```

The following example shows how to configure the device to discard path attributes in the range from 17 to 255 from incoming Update messages from the specified neighbor. The rest of the Update message will be processed as usual.

```
router bgp 65600
 neighbor 2001:DB8:1::1 path-attribute discard 17 255 in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
draft-ietf-idr-error-handling	Revised Error Handling for BGP Updates from External Neighbors

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling

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 [http://www.cisco.com/cisco/web/featurenavigator](#). An account on Cisco.com is not required.

Table 62: Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling

Feature Name	Releases	Feature Information
BGP Attribute Filter and Enhanced Attribute Error Handling	15.2(1)SY	<p>The BGP Attribute Filter allows you to “treat-as-withdraw” updates that contain specific path attributes. The prefixes contained in the update are removed from the routing table. The feature also allows you to remove specific path attributes from incoming updates. Both behaviors provide an increased measure of security. The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to errors from any malformed update, thereby saving resources.</p> <p>The following commands were introduced: bgp enhanced-error, neighbor path-attribute discard, neighbor path-attribute treat-as-withdraw, show ip bgp path-attribute discard, and show ip bgp path-attribute unknown.</p> <p>The following commands were modified: show ip bgp, show ip bgp neighbor, and show ip bgp vpnv4 all.</p>



CHAPTER 46

BGP—Support for iBGP Local-AS

Prior to the BGP—Support for iBGP Local-AS feature, the **neighbor local-as** command was used on a route reflector to customize AS_PATH attributes for routes received from an eBGP neighbor. The **neighbor local-as** command can now be used to enable the sending of the iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) over an iBGP local-AS session. This functionality is useful when merging two autonomous systems, when it is advantageous to keep the iBGP attributes in routes.

Prior to the BGP—Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes, providing more flexibility.

- [Finding Feature Information, page 773](#)
- [Restrictions for Support for iBGP Local-AS, page 774](#)
- [Information About Support for iBGP Local-AS, page 774](#)
- [Support for iBGP Local-AS, page 774](#)
- [Benefits of iBGP Local-AS, page 775](#)
- [How to Configure iBGP Local-AS, page 776](#)
- [Configuring iBGP Local-AS, page 776](#)
- [Configuration Examples for iBGP Local-AS, page 779](#)
- [Example: Configuring iBGP Local-AS, page 779](#)
- [Additional References for Support for iBGP Local-AS, page 780](#)
- [Feature Information for BGP—Support for iBGP Local-AS, page 780](#)

Finding Feature Information

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

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 Support for iBGP Local-AS

- This feature is not supported for a peer that belongs to a confederation.
- iBGP neighbors that are in a single AS are put into a separate update group from iBGP neighbors that are configured with the iBGP Local-AS feature.
- Two iBGP neighbors that are in two different autonomous systems and that are configured with the same iBGP Local-AS number are put into separate update groups.

Information About Support for iBGP Local-AS

Support for iBGP Local-AS

Prior to the Support for iBGP Local-AS feature, the **neighbor local-as** command was used to customize AS_PATH attributes for routes received from an eBGP neighbor. When a peer (or peer group) was configured with the **neighbor local-as** command and the **neighbor remote-as** command that specified the same AS number, that peer acted as an iBGP local-AS client, causing the local-AS number to be prepended to routes from that neighbor. However, the connection with the neighbor was not an iBGP connection, which meant that some attributes were not preserved in outgoing updates. Updates were propagated as if the peer were an eBGP peer, and therefore the LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST attributes were not added to the route during re-advertisement because those attributes apply to iBGP.

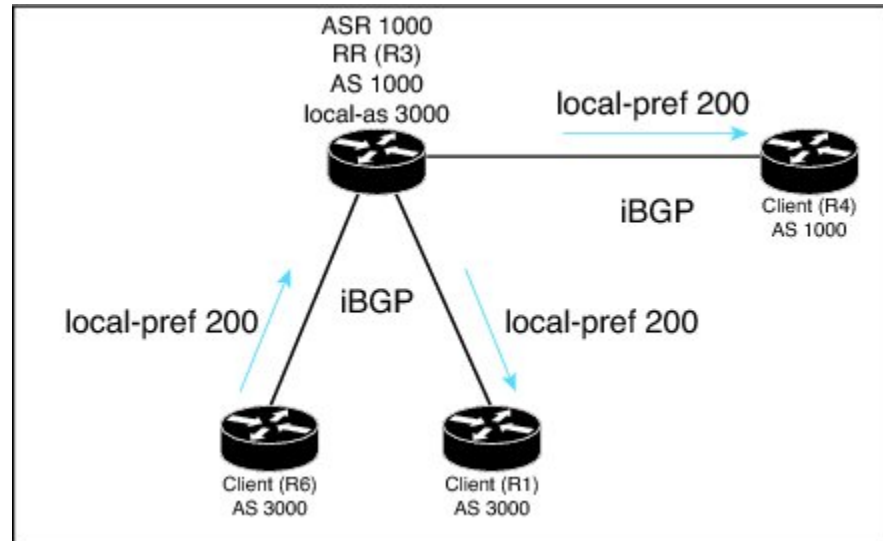
There can be scenarios where the network administrator wants those attributes to be passed unmodified, typically to facilitate the merging of two autonomous systems. The Support for iBGP Local-AS feature causes BGP to treat an iBGP local-AS connection as an iBGP connection and pass those four attributes unmodified.

In such a scenario, an RR is in one AS; the routers in the other AS become RR clients and keep their existing AS number. The RR is configured with the **neighbor remote-as** command and the **neighbor local-as** command that specify that neighbor (client) or peer group and its AS number. The session with that neighbor is treated as iBGP when the RR advertises routes received in the session to iBGP peers that have no **neighbor local-as** command configured. The RR transparently passes BGP updates to and from the client routers in the other AS, passing the LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST attributes unmodified instead of dropping them.

The figure below illustrates a scenario where this feature is being used to facilitate the merging of two autonomous systems. The route reflector (R3) and R4 belong to AS 1000; R1 and R6 belong to AS 3000. The RR is configured with the **neighbor local-as** command and the **neighbor remote-as** command, both specifying

AS 3000. Even though the routers belong to two different autonomous systems, the local preference value is preserved in the updates from the client (R6) to the RR, and out to the other clients.

Figure 67: Support for iBGP Local-AS to Preserve iBGP Policies Between Two Autonomous Systems



Prior to the Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes. This flexibility is achieved by configuring the **neighbor allow-policy** command on the RR, and configuring an outbound route map to send attributes downstream to peers.

Benefits of iBGP Local-AS

- One use case for this feature is when merging two ISPs that have different autonomous system numbers. It is desirable to preserve attributes that are considered internal (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) in the routes that are being passed to a different autonomous system. Configuring this feature causes these iBGP attributes in the advertised routes to be passed on to the second AS rather than dropped.
- Prior to the BGP—Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes. This flexibility is achieved by configuring the **neighbor allow-policy** command on the RR, and configuring an outbound route map to send iBGP attributes downstream to peers.

How to Configure iBGP Local-AS

Configuring iBGP Local-AS

Configure the iBGP Local-AS feature on a route reflector when you want iBGP attributes to be preserved in updates to a neighbor or peer group in another AS. Optionally, you can enable the route reflector to change iBGP attributes sent to a neighbor.



Note

This task includes the optional **neighbor allow-policy** command that enables you to configure the route reflector to change iBGP attributes, but it does not include the route map that actually changes iBGP attributes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **remote-as** *as-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **local-as** *as-number*
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **route-reflector-client**
10. **address-family** **vpn4**
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy**
12. **exit**
13. **address-family** **vpn6**
14. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy**
15. **end**
16. **show ip bgp vpn4 all neighbors** {*ip-address* | *ipv6-address*} **policy**
17. **show ip bgp vpn4 all update-group** *update-group*
18. **show ip bgp vpn4 all neighbors** {*ip-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Device(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 1000	Enters router configuration mode to create or configure a BGP routing process.
Step 5	neighbor <i>peer-group-name</i> peer-group Example: Device(config-router)# neighbor rr-client-ab peer-group	(Optional) Identifies a peer group.
Step 6	neighbor {<i>ip-address</i> <i>ipv6-address</i>} peer-group <i>peer-group-name</i> Example: Device(config-router)# neighbor 192.168.3.3 peer-group rr-client-ab	(Optional) Configures a BGP neighbor to be a member of a peer group.
Step 7	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor rr-client-ab remote-as 3000	Identifies the AS of the neighbor or peer group.
Step 8	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group</i>} local-as <i>as-number</i> Example: Device(config-router)# neighbor rr-client-ab local-as 3000	Configures the local-AS feature for the neighbor or peer group.

	Command or Action	Purpose
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group</i> } route-reflector-client Example: Device(config-router)# neighbor rr-client-ab route-reflector-client	Configures the local device to be a route reflector and configures the neighbor or peer group to be its client.
Step 10	address-family vpnv4 Example: Device(config-router)# address-family vpnv4	(Optional) Places the router in VPNv4 address family configuration mode.
Step 11	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group</i> } allow-policy Example: Device(config-router-af)# neighbor rr-client-ab allow-policy	(Optional) Allows the RR to be configured to change iBGP attributes for the specified neighbor or peer group.
Step 12	exit Example: Device(config-router-af)# exit	Exits address family configuration mode and enters router configuration mode.
Step 13	address-family vpnv6 Example: Device(config-router)# address-family vpnv6	(Optional) Places the router in VPNv6 address family configuration mode.
Step 14	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group</i> } allow-policy Example: Device(config-router-af)# neighbor rr-client-ab allow-policy	(Optional) Allows the RR to be configured to change iBGP attributes for the specified neighbor or peer group.
Step 15	end Example: Device(config-router-af)# end	Exits address family configuration mode, and enters privileged EXEC mode.
Step 16	show ip bgp vpnv4 all neighbors { <i>ip-address</i> <i>ipv6-address</i> } policy	(Optional) Displays the locally configured policies of the neighbor.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device# show ip bgp vpnv4 all neighbors 192.168.3.3 policy</pre>	<ul style="list-style-type: none"> The output includes the phrase “allow-policy” if the neighbor allow-policy command was configured for that neighbor.
Step 17	<p>show ip bgp vpnv4 all update-group <i>update-group</i></p> <p>Example:</p> <pre>Device# show ip bgp vpnv4 all update-group 2</pre>	<p>(Optional) Displays the information for the update group.</p> <ul style="list-style-type: none"> The output includes the phrase “Allow-policy” if the neighbor allow-policy command was configured for neighbors in the update group.
Step 18	<p>show ip bgp vpnv4 all neighbors {<i>ip-address</i> <i>ipv6-address</i>}</p> <p>Example:</p> <pre>Device# show ip bgp vpnv4 all neighbors 192.168.3.3</pre>	<p>(Optional) Displays information about the neighbor.</p> <ul style="list-style-type: none"> The output includes the remote AS and local AS, which will indicate the same AS number when the Support for iBGP Local-AS feature is configured.

Configuration Examples for iBGP Local-AS

Example: Configuring iBGP Local-AS

The example configures a route reflector (RR) in AS 4000 to treat BGP sessions with the peer group rr-client-2 in AS 2500 as iBGP sessions. That is, iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) will not be dropped from routes in advertisements to and from the peer group; the attributes will be passed unmodified. AS 2500 will be prepended to the AS_PATH attribute in routes to and from the peer group.

Additionally, the **neighbor allow-policy** command configures that the network administrator can configure iBGP policies on the RR. That is, an outbound route map can be configured to change attributes that are sent to the downstream peers. In this example, the feature is applied to VPNv4 and VPNv6 address families.

```
router bgp 4000
 neighbor rr-client-2 peer-group
 neighbor 192.168.1.1 peer-group rr-client-2
 neighbor 192.168.4.1 peer-group rr-client-2
 neighbor rr-client-2 remote-as 2500
 neighbor rr-client-2 local-as 2500
 neighbor rr-client-2 route-reflector-client
 address-family vpnv4
   neighbor rr-client-2 allow-policy
!
 address-family vpnv6
   neighbor rr-client-2 allow-policy
```

Additional References for Support for iBGP Local-AS

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Migration of autonomous systems	“BGP Support for Dual AS Configuration for Network AS Migrations” module in the <i>IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/support

Feature Information for BGP—Support for iBGP Local-AS

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 63: Feature Information for BGP—Support for iBGP Local-AS

Feature Name	Releases	Feature Information
BGP—Support for iBGP Local-AS	15.2(1)SY	<p>Prior to the BGP—Support for Local-AS feature, the neighbor local-as command was used on a route reflector to customize AS_PATH attributes for routes received from an eBGP neighbor. The neighbor local-as command can now be used to enable the sending of the iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) over an iBGP local-AS session. This functionality is useful when merging two autonomous systems, when it is advantageous to keep the iBGP attributes in routes.</p> <p>Prior to the BGP—Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes, providing more flexibility.</p> <p>The following command was introduced:</p> <ul style="list-style-type: none"> • neighbor allow-policy <p>The following commands were modified:</p> <ul style="list-style-type: none"> • neighbor local-as • show ip bgp vpnv4



BGP-Multiple Cluster IDs

The BGP—Multiple Cluster IDs feature allows an iBGP neighbor (usually a route reflector) to have multiple cluster IDs: a global cluster ID and additional cluster IDs that are assigned to clients (neighbors). Prior to the introduction of this feature, a device could have a single, global cluster ID.

When a network administrator configures per-neighbor cluster IDs:

- The loop prevention mechanism based on a `CLUSTER_LIST` is automatically modified to take into account multiple cluster IDs.
- A network administrator can disable client-to-client route reflection based on cluster ID.
- [Finding Feature Information, page 783](#)
- [Information About BGP-Multiple Cluster IDs, page 784](#)
- [How to Use BGP-Multiple Cluster IDs, page 787](#)
- [Configuration Examples for BGP-Multiple Cluster IDs, page 792](#)
- [Additional References, page 793](#)
- [Feature Information for BGP-Multiple Cluster IDs, page 794](#)

Finding Feature Information

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

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

Information About BGP-Multiple Cluster IDs

Benefit of Multiple Cluster IDs Per Route Reflector

The BGP—Multiple Cluster IDs feature allows a route reflector (RR) to belong to multiple clusters, and therefore have multiple cluster IDs. An RR can have a cluster ID configured on a global basis and a per-neighbor basis. A single cluster ID can be assigned to two or more iBGP neighbors. Prior to this feature, an RR had a single, global cluster ID, which was configured by the **bgp cluster-id** router configuration command.

When a cluster ID is configured per neighbor (by the **neighbor cluster-id** router configuration command), the following two changes occur:

- The loop prevention mechanism based on the CLUSTER_LIST attribute is automatically modified to take into account multiple cluster IDs.
- The network administrator can disable client-to-client route reflection based on cluster ID, which allows the network design to change.

The loop prevention mechanism and the CLUSTER_LIST propagation rules are described in the section “How a CLUSTER_LIST Attribute is Used.” Disabling client-to-client reflection is described in the section “Behaviors When Disabling Client-to-Client Route Reflection.”

How a CLUSTER_LIST Attribute is Used

The CLUSTER_LIST propagation rules differ among releases, depending on whether the device is running a Cisco software release generated before or after the BGP—Multiple Cluster IDs feature was implemented. The same is true for loop prevention based on the CLUSTER_LIST.

The CLUSTER_LIST behavior is described below. Classic refers to the behavior of software released before the multiple cluster IDs feature was implemented; MCID refers to the behavior of software released after the feature was implemented.

CLUSTER_LIST Propagation Rules

- Classic—Before reflecting a route, the RR appends the global cluster ID to the CLUSTER_LIST. If the received route had no CLUSTER_LIST attribute, the RR creates a new CLUSTER_LIST attribute with that global cluster ID.
- MCID—Before reflecting a route, the RR appends the cluster ID of the neighbor the route was received from to the CLUSTER_LIST. If the received route had no CLUSTER_LIST attribute, the RR creates a new CLUSTER_LIST attribute with that cluster ID. This behavior includes a neighbor that is not a client of the speaker. If the nonclient neighbor the route was received from does not have an associated cluster ID, the RR uses the global cluster ID.

Loop Prevention Based on CLUSTER_LIST

- Classic—When receiving a route, the RR discards the route if the RR's global cluster ID is contained in the CLUSTER_LIST of the route.
- MCID—When receiving a route, the RR discards the route if the RR's global cluster ID or any of the cluster IDs assigned to any of the iBGP neighbors is contained in the CLUSTER_LIST of the route.

Behaviors When Disabling Client-to-Client Route Reflection

With the introduction of multiple cluster IDs per iBGP neighbor, it is possible to disable route reflection from client to client on the basis of cluster ID. Disabling route reflection allows you to change the network design. A typical (but not required) scenario after disabling route reflection is that clients are fully meshed, so they have to send more updates, and the RR has client-to-client reflection disabled, so that it has to send fewer updates.

You might want to disable route reflection in a scenario similar to the one in the figure below. An RR has several clients [Provider-Edge (PE) routers] with which it has sessions. The iBGP neighbors that should belong to one cluster were assigned the same cluster ID.

Because the PEs belonging to the same cluster are fully meshed (PE1 and PE2 have a session between them; PE3 and PE4 have a session between them), there is no need to reflect the routes between them. That is, routes from PE1 should be forwarded to PE3 and PE4, but not to PE2.

It is important to know that when the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

Disabling client-to-client route reflection is done differently and has different results, depending on whether the device is running Cisco software generated before or after the multiple cluster IDs feature was implemented. Classic refers to the behavior of software released before the multiple cluster IDs feature was implemented; MCID refers to the behavior of software released after the multiple cluster IDs feature was implemented.

- Classic—When receiving a route from a client, the RR does not reflect it to any other client. Other scenarios for reflection (client-to-nonclient and nonclient-to-client) are maintained. Disabling of route reflection from client to client is usually done when all the clients are fully meshed (the routes are advertised between the clients via that mesh, so there is no need for reflection). The command to disable client-to-client route reflection is entered in router configuration mode (after the **router bgp** command) and it applies globally to all address families: **no bgp client-to-client reflection**
- MCID—When receiving a route from a client, the RR does not reflect it to another client if both clients belong to a cluster for which client-to-client reflection has been disabled. Therefore, route reflection is disabled only intracluster (within the cluster specified). Other cases for reflection (client-to-nonclient, nonclient-to-client, and intercluster) are maintained. This functionality is usually configured when all the clients for a particular cluster are fully meshed among themselves (but not with clients of other clusters). The command to disable client-to-client route reflection for a particular cluster is entered in router configuration mode and it applies globally to all address families:
no bgp client-to-client reflection intra-cluster cluster-id {any | cluster-id1 cluster-id2...}

The **any** keyword is used to disable client-to-client reflection for any cluster.

The Classic, previously released command for disabling all client-to-client reflection is also still available during this post-MCID release timeframe:

no bgp client-to-client reflection [all]

(The optional **all** keyword has no effect in either the positive or negative form of the command, and does not appear in configuration files. It is just to remind the network administrator that both intercluster and intracluster client-to-client reflection are enabled or disabled.)

In summary, after the introduction of the multiple cluster IDs feature, there are three levels of configuration that can disable client-to-client reflection. The software performs them in the following order, from least specific to most specific:

- 1 Least specific: **no bgp client-to-client reflection [all]** Disables intracluster and intercluster client-to-client reflection.
- 2 More specific: **no bgp client-to-client reflection intra-cluster cluster-id any** Disables intracluster client-to-client reflection for any cluster-id.
- 3 Most specific: **no bgp client-to-client reflection intra-cluster cluster-id *cluster-id1 cluster-id2 ...*** Disables intracluster client-to-client reflection for the specified clusters.

When BGP is advertising updates, the software evaluates each level of configuration in order. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is necessary.

Note the results of the base (positive) and negative (**no**) forms of the three commands listed above:

- A negative configuration (that is, with the **no** keyword) overwrites any less specific configuration.
- A positive configuration (that is, without the **no** keyword) will lose out to (default to) what is configured in a less specific configuration.
- Configurations at any level appear in the configuration file only if they are negative.

All levels can be configured independently and all levels appear in the configuration file independently of the configuration of other levels.

Note that negative configuration makes any more specific configuration unnecessary (because even if the more specific configuration is positive, it is not processed after the negative configuration; if the more specific configuration is negative, it is functionally the same as the earlier negative configuration). The following examples illustrate this behavior.

Example 1

no bgp client-to-client reflection

no bgp client-to-client reflection intra-cluster cluster-id any

Intercluster and intracluster reflection are disabled (based on the first command). The second command disables intracluster reflection, but it is unnecessary because intracluster reflection is already disabled by the first command.

Example 2

no bgp client-to-client reflection intra-cluster cluster-id any

bgp client-to-client reflection intra-cluster cluster-id 1.1.1.1

Cluster ID 1.1.1.1 has intracluster route reflection disabled (even though the second command is positive), because the first command is used to evaluate the update. The first command was negative, and once any level of configuration disables client-to-client reflection, no further evaluation is performed.

Another way to look at this example is that the second command, because it is in a positive form, defaults to the behavior of the first command (which is less specific). Thus, the second command is unnecessary.

Note that the second command would not appear in a configuration file because it is not a negative command.

How to Use BGP-Multiple Cluster IDs

Configuring a Cluster ID per Neighbor

Perform this task on an iBGP peer (usually a route reflector) to configure a cluster ID per neighbor. Configuring a cluster ID per neighbor causes the loop-prevention mechanism based on the CLUSTER_LIST to be automatically modified to take into account multiple cluster IDs. Also, you gain the ability to disable client-to-client route reflection on the basis of cluster ID. The software tags the neighbor so that you can disable route reflection with the use of another command. (See the tasks for disabling client-to-client reflection later in this module.)



Note

When you change a cluster ID for a neighbor, BGP automatically does an inbound soft refresh and an outbound soft refresh for all iBGP peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number*
5. **neighbor** {*ip-address* | *ipv6-address*} **cluster-id** *cluster-id*
6. **end**
7. **show ip bgp cluster-ids**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to

	Command or Action	Purpose
		65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	neighbor <i>{ip-address ipv6-address}</i> remote-as <i>autonomous-system-number</i> Example: <pre>Device(config-router)# neighbor 192.168.1.2 remote-as 65000</pre>	Adds an entry to the BGP routing table.
Step 5	neighbor <i>{ip-address ipv6-address}</i> cluster-id <i>cluster-id</i> Example: <pre>Device(config-router)# neighbor 192.168.1.2 cluster-id 0.0.0.1</pre>	Assigns a cluster ID to the specified neighbor. <ul style="list-style-type: none"> The cluster ID can be in dotted decimal format (such as 192.168.7.4) or decimal format (such as 23), with a maximum of 4 bytes. A cluster ID that is configured in decimal format (such as 23) is modified to dotted decimal format (such as 0.0.0.23) when it appears in a configuration file. When you change a cluster ID for a neighbor, BGP automatically does an inbound soft refresh and an outbound soft refresh for all iBGP peers.
Step 6	end Example: <pre>Device(config-router)# end</pre>	(Optional) Exits to privileged EXEC mode.
Step 7	show ip bgp cluster-ids Example: <pre>Device# show ip bgp cluster-ids</pre>	(Optional) Lists: <ul style="list-style-type: none"> the global cluster ID (whether configured or not) all cluster IDs that are configured to a neighbor all cluster IDs for which the network administrator has disabled reflection

Disabling Intracluster and Intercluster Client-to-Client Reflection

Perform the following task on a route reflector if you want to disable both intracluster and intercluster client-to-client reflection. Doing so is the broadest (least specific) way to disable client-to-client reflection. Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.



Note When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **no bgp client-to-client reflection [all]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	no bgp client-to-client reflection [all] Example: Device(config-router)# no bgp client-to-client reflection all	Disables intracluster and intercluster client-to-client route reflection. <ul style="list-style-type: none"> • The all keyword is just to emphasize that the bgp client-to-client reflection command affects both intracluster and intercluster reflection; the all keyword has no effect in the positive or negative form of the command.

Disabling Intracluster Client-to-Client Reflection for Any Cluster ID

Perform the following task on a route reflector to disable intracluster client-to-client reflection for any cluster ID. Doing so is considered to be the middle of the three levels of commands available to disable client-to-client

reflection. That is, it is more specific than disabling intracluster and intercluster client-to-client reflection, but it is not as specific as disabling intracluster client-to-client reflection for certain cluster IDs.

Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.



Note When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **no bgp client-to-client reflection intra-cluster cluster-id any**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	no bgp client-to-client reflection intra-cluster cluster-id any Example: Device(config-router)# no bgp	Disables intracluster client-to-client route reflection for any cluster.

	Command or Action	Purpose
	<code>client-to-client reflection intra-cluster cluster-id any</code>	

Disabling Intracluster Client-to-Client Reflection for Specified Cluster IDs

Perform the following task on a route reflector to disable intracluster client-to-client reflection for specified cluster IDs. Doing so is considered to be the most specific of the three levels of commands available to disable client-to-client reflection. Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.



Note

When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp as-number`
4. `no bgp client-to-client reflection intra-cluster cluster-id cluster-id1 [cluster-id2...]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p><code>router bgp <i>as-number</i></code></p> <p>Example:</p> <pre>Device(config)# router bgp 65000</pre>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to

	Command or Action	Purpose
		65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	<p>no bgp client-to-client reflection intra-cluster cluster-id <i>cluster-id1</i> [<i>cluster-id2...</i>]</p> <p>Example:</p> <pre>Device(config-router)# no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 0.0.0.3 105</pre>	<p>Disables intracluster client-to-client route reflection within each of the specified clusters.</p> <ul style="list-style-type: none"> Note that this example command will appear in the configuration file as “no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 0.0.0.3 0.0.0.105” because decimal cluster ID numbers appear in the dotted decimal format.

Configuration Examples for BGP-Multiple Cluster IDs

Example: Per-Neighbor Cluster ID

The following example is configured on a route reflector. The neighbor (client) at IPv6 address 2001:DB8:1::1 is configured to have the cluster ID of 0.0.0.6:

```
router bgp 6500
 neighbor 2001:DB8:1::1 cluster-id 0.0.0.6
```

Example: Disabling Client-to-Client Reflection

The following example disables all intracluster and intercluster client-to-client reflection:

```
router bgp 65000
 no bgp client-to-client reflection all
```

The following example disables intracluster client-to-client reflection for any cluster ID:

```
router bgp 65000
 no bgp client-to-client reflection intra-cluster cluster-id any
```

The following example disables intracluster client-to-client reflection for the specified cluster IDs 0.0.0.1, 14, 15, and 0.0.0.6:

```
router bgp 65000
 no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 14 15 0.0.0.6
```

Remember that a cluster ID specified in the **neighbor cluster-id** command in decimal format (such as 23) will appear in a configuration file in dotted decimal format (such as 0.0.0.23). The decimal format does not appear in the configuration file. The running configuration might look like this:

```
router bgp 65000
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.6
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.14
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.15
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP-Multiple Cluster IDs

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 64: Feature Information for BGP—Multiple Cluster IDs

Feature Name	Releases	Feature Information
BGP—Multiple Cluster IDs	15.2(1)SY	<p>The BGP—Multiple Cluster IDs feature allows an iBGP neighbor (usually a route reflector) to have multiple cluster IDs: a global cluster ID and additional cluster IDs that are assigned to clients (neighbors). Prior to the introduction of this feature, a device could have a single, global cluster ID.</p> <p>When a network administrator configures per-neighbor cluster IDs:</p> <ul style="list-style-type: none"> • The loop prevention mechanism based on a <code>CLUSTER_LIST</code> is automatically modified to take into account multiple cluster IDs. • A network administrator can disable client-to-client route reflection based on cluster ID. <p>The following commands were introduced:</p> <ul style="list-style-type: none"> • bgp client-to-client reflection intra-cluster • neighbor cluster-id • show ip bgp cluster-ids <p>The following commands were modified:</p> <ul style="list-style-type: none"> • bgp client-to-client reflection • show ip bgp neighbors • show ip bgp template peer-session • show ip bgp update-group



CHAPTER 48

BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

The BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard feature introduces the ability to set a range of route target (RT) community attributes or VPN distinguisher community attributes when mapping them. A network administrator might want to map one or more RTs at an egress ASBR to different RTs at an ingress ASBR. The VPN Distinguisher Attribute feature allows an administrator to map RTs to a VPN distinguisher that is carried through an eBGP and then mapped to RTs at an ingress ASBR. The mapping is achieved by configuring a route map that sets an RT range or VPN distinguisher range of extended community attributes. Specifying a range rather than individual RTs saves time and simplifies the configuration. Furthermore, a VPN distinguisher range allows more than one VPN distinguisher attribute per route-map clause, thereby removing the restriction that applied prior to this feature.

- [Finding Feature Information, page 797](#)
- [Restrictions for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard, page 798](#)
- [Information About BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard, page 798](#)
- [How to Map RTs to RTs Using a Range, page 798](#)
- [Configuration Examples for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard, page 805](#)
- [Additional References for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard, page 807](#)
- [Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard, page 807](#)

Finding Feature Information

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

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

Restrictions for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

- A range (specified in the **set extcommunity rt** command or the **set extcommunity vpn-distinguisher** command) can include a maximum of 450 extended communities.
- The VPN distinguisher range is not relayed to an iBGP peer.

Information About BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Benefits of RT and VPN Distinguisher Attribute Mapping Range

A network administrator might want to rewrite (or map) one or more route targets (RTs) at an egress ASBR to different RTs at an ingress ASBR. One use case would be to keep the RTs at the egress ASBR private from the ingress ASBR.

The rewrite is achieved by using inbound route maps, matching prefixes to route-map clauses that match inbound RTs, and mapping those RTs to different RTs recognized by the neighbor AS. Such a rewrite configuration could be complex on inbound route maps, with potentially hundreds of RTs that would need to be specified individually (configuring **set extcommunity rt value1 value2 value3 ...**). If the RTs being attached to the prefixes are consecutive, the configuration can be simplified by specifying a range of RTs. Thus, the benefits of the RT mapping range are saving time and simplifying the configuration.

Likewise, the mapping of RTs to a VPN distinguisher attribute (and vice versa) can also be simplified by specifying a range of RTs or VPN distinguishers. The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source RTs private from an ASBR in a destination AS. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.

The RT and VPN Distinguisher Attribute Mapping Range feature introduces the ability to specify a range of either route targets (RTs) or VPN distinguishers when mapping them.

Another benefit applies to setting a VPN distinguisher. Prior to this feature, only one **set extcommunity vpn-distinguisher** value was allowed per route-map clause. With the introduction of the mapping range, a range of VPN distinguishers can be set on a route.

How to Map RTs to RTs Using a Range

Replacing an RT with a Range of RTs

Perform this task on an egress ASBR to replace a route target (RT) with an RT range. Remember to replace the range of RTs with an RT on the ingress ASBR; that task is described in the “Replacing a Range of RTs with an RT” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} *rt value*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity** *rt range* *start-value* *end-value*
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family** **vpn4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-tag* **out**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip extcommunity-list <i>expanded-list</i> { permit deny } <i>rt value</i> Example: Device(config)# ip extcommunity-list 22 permit rt 101:100	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT are in the extended community list. <ul style="list-style-type: none"> • This example permits routes having RT 101:100 into the extended community list 22.

	Command or Action	Purpose
Step 4	exit Example: Device(config-extcomm-list)# exit	Exits the configuration mode and enters the next higher configuration mode.
Step 5	route-map map-tag {permit deny} <i>[sequence-number]</i> Example: Device(config)# route-map rt-mapping permit 10	Configures a route map that permits or denies the routes allowed by the subsequent match command. <ul style="list-style-type: none"> • This example permits the routes allowed by the subsequent match command.
Step 6	match extcommunity <i>extended-community-list-name</i> Example: Device(config-route-map)# match extcommunity 22	Matches on the specified community list. <ul style="list-style-type: none"> • For this example, routes that match the extended community list 22 (which was configured in Step 3) are subject to the subsequent set commands.
Step 7	set extcomm-list extcommunity-name delete Example: Device(config-route-map)# set extcomm-list 22 delete	Deletes the RT from routes that are in the specified extended community list. <ul style="list-style-type: none"> • For this example, RTs are deleted from routes that are in extended community list 22.
Step 8	set extcommunity rt range start-value end-value Example: Device(config-route-map)# set extcommunity rt range 500:1 500:9	For the routes that are permitted by the route map, sets the specified RT range of extended community attributes, inclusive. <ul style="list-style-type: none"> • For this example, routes that match extended community 22 have their RT extended community attribute values set to 500:1, 500:2, 500:3, 500:4, 500:5, 500:6, 500:7, 500:8, and 500:9.
Step 9	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 10	route-map map-tag {permit deny} <i>[sequence-number]</i> Example: Device(config)# route-map rt-mapping permit 20	(Optional) Configures a route map entry that permits routes. <ul style="list-style-type: none"> • This example configures a route map entry that permits other routes not subject to the RT-to-RT range mapping. If you do not perform this step, all other routes are subject to an implicit deny.

	Command or Action	Purpose
Step 11	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>as-number</i> Example: Device(config)# router bgp 3000	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor <i>ip-address remote-as autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.103.1 remote-as 3000	Specifies that the neighbor belongs to the autonomous system.
Step 14	address-family <i>vpn4</i> Example: Device(config-router)# address-family vpn4	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	neighbor <i>ip-address activate</i> Example: Device(config-router-af)# neighbor 192.168.103.1 activate	Activates the specified neighbor.
Step 16	neighbor <i>ip-address route-map map-tag out</i> Example: Device(config-router-af)# neighbor 192.168.103.1 route-map rt-mapping out	Applies the specified outgoing route map to the specified neighbor.
Step 17	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and enters privileged EXEC mode.

Replacing a Range of RTs with an RT

Perform this task on an ingress ASBR to replace an RT range of attributes with an RT attribute. This task assumes you already configured the egress ASBR to replace the RT with an RT range; that task is described in the “Replacing an RT with a Range of RTs” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *reg-exp*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity** **rt** *value* **additive**
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family** **vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-tag* **in**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>ip extcommunity-list <i>expanded-list</i> {permit deny} rt <i>reg-exp</i></p> <p>Example:</p> <pre>Device(config)# ip extcommunity-list 128 permit rt 500:[1-9]</pre>	<p>Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT range are in the extended community list.</p> <ul style="list-style-type: none"> This example permits routes having RTs in the range 500:1 to 500:9 into the extended community list 128.
Step 4	<p>exit</p> <p>Example:</p> <pre>Device(config-extcomm-list)# exit</pre>	<p>Exits the configuration mode and enters the next higher configuration mode.</p>
Step 5	<p>route-map <i>map-tag</i> {permit deny} [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map rtm2 permit 10</pre>	<p>Configures a route map that permits or denies the routes allowed by the subsequent match command.</p> <ul style="list-style-type: none"> This example permits the routes allowed by the subsequent match command.
Step 6	<p>match extcommunity <i>extended-community-list-name</i></p> <p>Example:</p> <pre>Device(config-route-map)# match extcommunity 128</pre>	<p>Matches on the specified community list.</p> <ul style="list-style-type: none"> In this example, routes that match the extended community list 128 (which was configured in Step 3) are subject to the subsequent set commands.
Step 7	<p>set extcomm-list <i>extcommunity-name</i> delete</p> <p>Example:</p> <pre>Device(config-route-map)# set extcomm-list 128 delete</pre>	<p>Deletes the RTs in the range from routes that are in the specified extended community list.</p> <ul style="list-style-type: none"> In this example, RTs in the range are deleted from routes that are in extended community list 128.
Step 8	<p>set extcommunity rt <i>value</i> additive</p> <p>Example:</p> <pre>Device(config-route-map)# set extcommunity rt 400:1 additive</pre>	<p>Sets the routes that are permitted by the route map with the specified RT.</p> <ul style="list-style-type: none"> In this example, routes that match extended community 128 have their RT set to 400:1. The additive keyword causes the RT to be added to the RT list without replacing any RTs.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route-map configuration mode and enters global configuration mode.</p>

	Command or Action	Purpose
Step 10	route-map <i>map-tag</i> { permit deny } [<i>sequence-number</i>] Example: Device(config)# route-map rmap2 permit 20	(Optional) Configures a route map entry that permits routes. <ul style="list-style-type: none"> This example configures a route map entry that permits other routes not subject to the RT-range-to-RT mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>as-number</i> Example: Device(config)# router bgp 4000	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.0.50 remote-as 4000	Specifies that the neighbor belongs to the autonomous system.
Step 14	address-family vpn4 Example: Device(config-router-af)# address-family vpn4	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 15	neighbor <i>ip-address</i> activate Example: Device(config-router-af)# neighbor 192.168.0.50 activate	Activates the specified neighbor.
Step 16	neighbor <i>ip-address</i> route-map <i>map-tag</i> in Example: Device(config-router-af)# neighbor 192.168.0.50 route-map rmap2 in	Applies the specified incoming route map to the specified neighbor.

	Command or Action	Purpose
Step 17	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and enters privileged EXEC mode.

Configuration Examples for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Example: Replacing an RT with a Range of RTs

In the following example, on the egress ASBR, routes having RT 101:100 are in the extended community list 22. A route-map named rt-mapping matches on extended community list 22 and deletes the RT from routes in the community list. Routes that match the community list have their RT set to an RT in the range from 500:1 to 500:9. The route map is applied to the neighbor 192.168.103.1.

Egress ASBR

```
ip extcommunity-list 22 permit rt 101:100
!
route-map rt-mapping permit 10
  match extcommunity 22
  set extcomm-list 22 delete
  set extcommunity rt range 500:1 500:9
!
route-map rt-mapping permit 20
!
router bgp 3000
  neighbor 192.168.103.1 remote-as 3000
  address-family vpnv4
    neighbor 192.168.103.1 activate
    neighbor 192.168.103.1 route-map rt-mapping out
  exit-address-family
!
```

On the ingress ASBR, RTs in the range 500:1 to 500:9 belong to extended community list 128. A route map named rtm2 maps those RTs to RT 400:1. The route map is applied to the neighbor 192.168.0.50.

Ingress ASBR

```
ip extcommunity-list 128 permit RT:500:[1-9]
!
route-map rtm2 permit 10
  match extcommunity 128
  set extcomm-list 128 delete
  set extcommunity rt 400:1 additive
!
route-map rtm2 permit 20
!
router bgp 4000
  neighbor 192.168.0.50 remote-as 4000
```

Example: Replacing an RT with a Range of VPN Distinguishers

```

address-family vpnv4
 neighbor 192.168.0.50 activate
 neighbor 192.168.0.50 route-map rmap2 in
 exit-address-family
!
```

Example: Replacing an RT with a Range of VPN Distinguishers

In the following example, on the egress ASBR, routes having RT 201:100 are in the extended community list 22. A route-map named rt-mapping matches on extended community list 22 and deletes the RT from routes in the community list. Routes that match the community list have their VPN distinguishers set to VPN distinguishers in the range from 600:1 to 600:8. The route map is applied to the neighbor 192.168.103.1.

Egress ASBR

```

ip extcommunity-list 22 permit rt 201:100
!
route-map rt-mapping permit 10
 match extcommunity 22
 set extcomm-list 22 delete
 set extcommunity vpn-distinguisher range 600:1 600:8
!
route-map rt-mapping permit 20
!
router bgp 3000
 neighbor 192.168.103.1 remote-as 3000
 address-family vpnv4
  neighbor 192.168.103.1 activate
  neighbor 192.168.103.1 route-map rt-mapping out
 exit-address-family
!
```

On the ingress ASBR, VPN distinguishers in the range 600:1 to 600:8 belong to extended community list 101. A route map named rmap2 maps those VPN distinguishers to RT range 700:1 700:10. The route map is applied to the neighbor 192.168.0.50. The additive option adds the new range to the existing value without replacing it.

Ingress ASBR

```

ip extcommunity-list 101 permit VD:600:[1-8]
!
route-map rmap2 permit 10
 match extcommunity 101
 set extcomm-list 101 delete
 set extcommunity rt 700:1 700:10 additive
!
route-map rmap2 permit 20
!
router bgp 4000
 neighbor 192.168.0.50 remote-as 4000
 address-family vpnv4
  neighbor 192.168.0.50 activate
  neighbor 192.168.0.50 route-map rmap2 in
 exit-address-family
!
```


Additional References for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP—VPN Distinguisher Attribute	“BGP—VPN Distinguisher Attribute” module in the <i>IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/support

Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 65: Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Feature Name	Releases	Feature Information
BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard	15.2(1)SY	<p>The BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard feature introduces the ability to set a range of route target (RT) community attributes or VPN distinguisher community attributes when mapping them. A network administrator might want to map one or more RTs at an egress ASBR to different RTs at an ingress ASBR. The VPN Distinguisher Attribute feature allows an administrator to map RTs to a VPN distinguisher that is carried through an eBGP and then mapped to RTs at an ingress ASBR. The mapping is achieved by configuring a route map that sets an RT range or VPN distinguisher range of extended community attributes. Specifying a range rather than individual RTs saves time and simplifies the configuration. Furthermore, a VPN distinguisher range allows more than one VPN distinguisher attribute per route-map clause, thereby removing the restriction that applied prior to this feature.</p> <p>The following commands were modified:</p> <ul style="list-style-type: none"> • set extcommunity rt • set extcommunity vpn-distinguisher



CHAPTER 49

BGP-VPN Distinguisher Attribute

The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source route targets (RTs) private from an Autonomous System Border Router (ASBR) in a destination autonomous system. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.

- [Finding Feature Information, page 809](#)
- [Information About BGP-VPN Distinguisher Attribute, page 809](#)
- [How to Configure BGP-VPN Distinguisher Attribute, page 811](#)
- [Configuration Examples for BGP-VPN Distinguisher Attribute, page 818](#)
- [Additional References, page 819](#)
- [Feature Information for BGP-VPN Distinguisher Attribute, page 820](#)

Finding Feature Information

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

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

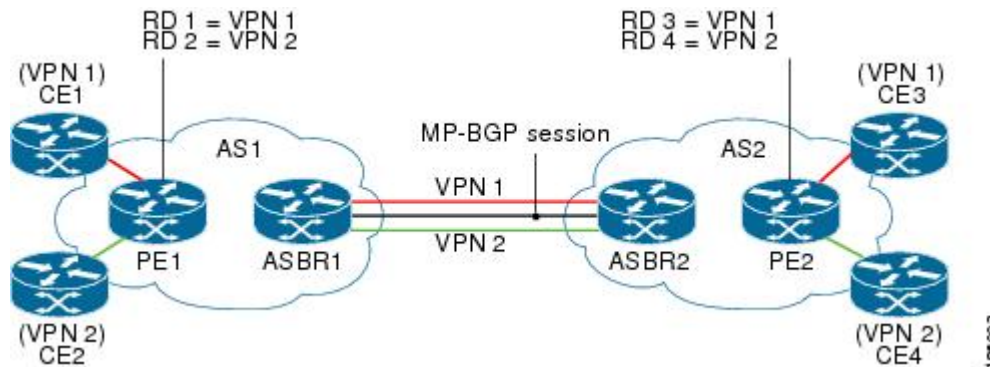
Information About BGP-VPN Distinguisher Attribute

Role and Benefit of the VPN Distinguisher Attribute

Route-target (RT) extended community attributes identify the VPN membership of routes. The RT attributes are placed onto a route at the exporting (egress) provider edge router (PE) and are transported across the iBGP cloud and across autonomous systems. Any Virtual Routing and Forwarding (VRF) instances at the remote PE that want to import such routes must have the corresponding RTs set as import RTs for that VRF.

The figure below illustrates two autonomous systems, each containing customer edge routers (CEs) that belong to different VPNs. Each PE tracks which route distinguisher (RD) corresponds to which VPN, thus controlling the traffic that belongs to each VPN.

Figure 69: Scenario in Which ASBRs Translate RTs Between Autonomous Systems



In an Inter-AS Option B scenario like the one in the figure above, these routes are carried across an AS boundary from Autonomous System Border Router 1 (ASBR1) to ASBR2 over an MP-eBGP session, with the routes' respective RTs as extended community attributes being received by ASBR2.

ASBR2 must maintain complex RT mapping schemes to translate RTs originated by AS1 to RTs recognized by AS2, so that the RTs can be imported by their respective VPN membership CE connections on PE2 for CE3 and CE4.

Some network administrators prefer to hide the RTs they source in AS1 from devices in AS2. In order to do that, the administrator must differentiate routes belonging to each VPN with a certain attribute so that the RTs can be removed on the outbound side of ASBR1 before sending routes to ASBR2, and ASBR2 can then map that attribute to recognizable RTs in AS2. The VPN Distinguisher (VD) extended community attribute serves that purpose.

The benefit of the BGP—VPN Distinguisher Attribute feature is that source RTs can be kept private from devices in destination autonomous systems.

How the VPN Distinguisher Attribute Works

The network administrator configures the egress ASBR to perform translation of RTs to a VPN distinguisher extended community attribute, and configures the ingress ASBR to perform translation of the VPN distinguisher to RTs. More specifically, the translation is achieved as follows:

On the Egress ASBR

- An outbound route map specifies a **match extcommunity** clause that determines which VPN routes are subject to mapping, based on the route's RT values.
- A **set extcommunity vpn-distinguisher** command sets the VPN distinguisher that replaces the RTs.
- The **set extcomm-list delete** command that references the same set of RTs is configured to remove the RTs, and then the route is sent to the neighboring ingress ASBR.

On the Ingress ARBR

- An inbound route map specifies a **match extcommunity vpn-distinguisher** command that determines which VPN routes are subject to mapping, based on the route's VPN distinguisher.
- The **set extcommunity rt** command specifies the RTs that replace the VPN distinguisher.
- For routes that match the clause, the VPN distinguisher is replaced with the configured RTs.

Additional Behaviors Related to the VPN Distinguisher

On the egress ASBR, if a VPN route matches a route map clause that does not have the **set extcommunity vpn-distinguisher** command configured, the RTs that the VPN route is tagged with are retained.

The VPN distinguisher is transitive across the AS boundary, but is not carried within the iBGP cloud. That is, the ingress ASBR can receive the VPN distinguisher from an eBGP peer, but the VPN distinguisher is discarded on the inbound side after it is mapped to the corresponding RTs.

On the ingress ASBR, if a VPN route carrying the VPN distinguisher matches a route map clause that does not have a **set extcommunity rt** command configured in the inbound route map, the system does not discard the attribute, nor does it propagate the attribute within the iBGP cloud. The VPN distinguisher for the route is retained so that the network administrator can configure the correct inbound policy to translate the VPN distinguisher to the RTs that the VPN route should carry. If the route is sent to eBGP peers, the VPN distinguisher is carried as is. The network administrator could configure a route-map entry to remove the VPN distinguisher from routes sent to eBGP peers.

Configuring a **set extcommunity vpn-distinguisher** command in an outbound route map or a **match extcommunity** command in an inbound route map results in an outbound or inbound route refresh request, respectively, in order to update the routes being sent or received.

How to Configure BGP-VPN Distinguisher Attribute

Replacing an RT with a VPN Distinguisher Attribute

Perform this task on an egress ASBR to replace a route target (RT) with a VPN distinguisher extended community attribute. Remember to replace the VPN distinguisher with a route target on the ingress ASBR; that task is described in the “Replacing a VPN Distinguisher Attribute with an RT” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} *rt value*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity vpn-distinguisher** *id*
9. **exit**
10. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor ip-address remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor ip-address activate**
16. **neighbor ip-address route-map** *map-name* **out**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip extcommunity-list <i>expanded-list</i> { permit deny } rt value Example: Device(config)# ip extcommunity-list 4 permit rt 101:100	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT are in the extended community list. <ul style="list-style-type: none"> • This example permits routes having RT 101:100 into the extended community list 4.

	Command or Action	Purpose
Step 4	exit Example: Device(config-extcomm-list)# exit	Exits the configuration mode and enters the next higher configuration mode.
Step 5	route-map map-tag {permit deny} <i>[sequence-number]</i> Example: Device(config)# route-map vpn-id-map1 permit 10	Configures a route map that permits or denies the routes allowed by the subsequent match command. <ul style="list-style-type: none"> • This example permits the routes allowed by the subsequent match command.
Step 6	match extcommunity <i>extended-community-list-name</i> Example: Device(config-route-map)# match extcommunity 4	Matches on the specified community list. <ul style="list-style-type: none"> • For this example, routes that match the extended community list 4 (which was configured in Step 3) are subject to the subsequent set commands.
Step 7	set extcomm-list extcommunity-name delete Example: Device(config-route-map)# set extcomm-list 4 delete	Deletes the RT from routes that are in the specified extended community list. <ul style="list-style-type: none"> • For this example, RTs are deleted from routes that are in extended community list 4.
Step 8	set extcommunity vpn-distinguisher id Example: Device(config-route-map)# set extcommunity vpn-distinguisher 111:100	For the routes that are permitted by the route map, sets the specified VPN distinguisher. <ul style="list-style-type: none"> • For this example, routes that match extended community 4 have their VPN distinguisher set to 111:100.
Step 9	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 10	route-map map-name {permit deny} <i>[sequence-number]</i> Example: Device(config)# route-map vpn-id-map1 permit 20	(Optional) Configures a route map entry that permits routes. <ul style="list-style-type: none"> • This example configures a route map entry that permits other routes not subject to the RT-to-VPN distinguisher mapping. If you do not perform this step, all other routes are subject to an implicit deny.

	Command or Action	Purpose
Step 11	exit Example: Device(config-route-map) # exit	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>as-number</i> Example: Device(config) # router bgp 2000	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router) # neighbor 192.168.101.1 remote-as 2000	Specifies that the neighbor belongs to the autonomous system.
Step 14	address-family <i>vpn4</i> Example: Device(config-router) # address-family vpn4	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	neighbor <i>ip-address</i> activate Example: Device(config-router-af) # neighbor 192.168.101.1 activate	Activates the specified neighbor.
Step 16	neighbor <i>ip-address</i> route-map <i>map-name</i> out Example: Device(config-router-af) # neighbor 192.168.101.1 route-map vpn-id-map1 out	Applies the specified outgoing route map to the specified neighbor.
Step 17	exit-address-family Example: Device(config-router-af) # exit-address-family	Exits address family configuration mode and enters privileged EXEC mode.

Replacing a VPN Distinguisher Attribute with an RT

Perform this task on an ingress ASBR to replace a VPN distinguisher extended community attribute with a route target (RT) attribute. This task assumes you already configured the egress ASBR to replace the RT with a VPN distinguisher; that task is described in the “Replacing an RT with a VPN Distinguisher Attribute” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **vpn-distinguisher** *id*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity** **rt** *value* **additive**
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family** **vpn4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-name* **in**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>ip extcommunity-list <i>expanded-list</i> {permit deny} vpn-distinguisher <i>id</i></p> <p>Example:</p> <pre>Device(config)# ip extcommunity-list 51 permit vpn-distinguisher 111:100</pre>	<p>Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified VPN distinguisher are in the extended community list.</p> <ul style="list-style-type: none"> This example permits routes having VPN distinguisher 111:110 into the extended community list 51.
Step 4	<p>exit</p> <p>Example:</p> <pre>Device(config-extcomm-list)# exit</pre>	<p>Exits the configuration mode and enters the next higher configuration mode.</p>
Step 5	<p>route-map <i>map-tag</i> {permit deny} [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map vpn-id-rewrite-map1 permit 10</pre>	<p>Configures a route map that permits or denies the routes allowed by the subsequent match command.</p> <ul style="list-style-type: none"> This example permits the routes allowed by the subsequent match command.
Step 6	<p>match extcommunity <i>extended-community-list-name</i></p> <p>Example:</p> <pre>Device(config-route-map)# match extcommunity 51</pre>	<p>Matches on the specified community list.</p> <ul style="list-style-type: none"> For this example, routes that match the extended community list 51 (which was configured in Step 3) are subject to the subsequent set commands.
Step 7	<p>set extcomm-list <i>extcommunity-name</i> delete</p> <p>Example:</p> <pre>Device(config-route-map)# set extcomm-list 51 delete</pre>	<p>Deletes the VPN distinguisher from routes that are in the specified extended community list.</p> <ul style="list-style-type: none"> For this example, VPN distinguishers are deleted from routes that are in extended community list 51.
Step 8	<p>set extcommunity <i>rt value</i> additive</p> <p>Example:</p> <pre>Device(config-route-map)# set extcommunity rt 101:1 additive</pre>	<p>Sets the routes that are permitted by the route map with the specified RT.</p> <ul style="list-style-type: none"> For this example, routes that match extended community 51 have their RT set to 101:1. The additive keyword causes the RT to be added to the RT list without replacing any RTs.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	<p>Exits route-map configuration mode and enters global configuration mode.</p>

	Command or Action	Purpose
Step 10	route-map <i>map-tag</i> { permit deny } [<i>sequence-number</i>] Example: Device(config)# route-map vpn-id-rewrite-map1 permit 20	(Optional) Configures a route map entry that permits routes. <ul style="list-style-type: none"> This example configures a route map entry that permits other routes not subject to the VPN distinguisher-to-RT mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>as-number</i> Example: Device(config)# router bgp 3000	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.0.81 remote-as 3000	Specifies that the neighbor belongs to the autonomous system.
Step 14	address-family vpn4 Example: Device(config-router-af)# address-family vpn4	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	neighbor <i>ip-address</i> activate Example: Device(config-router-af)# neighbor 192.168.0.81 activate	Activates the specified neighbor.
Step 16	neighbor <i>ip-address</i> route-map <i>map-name</i> in Example: Device(config-router-af)# neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in	Applies the specified outgoing route map to the specified neighbor.

	Command or Action	Purpose
Step 17	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and enters privileged EXEC mode.

Configuration Examples for BGP-VPN Distinguisher Attribute

Example: Translating RT to VPN Distinguisher to RT

The following example shows the egress ASBR configuration to replace a route target (RT) with a VPN distinguisher, and shows the ingress ASBR configuration to replace the VPN distinguisher with a route target.

On the egress ASBR, IP extended community list 1 is configured to filter VPN routes by permitting only routes with RT 101:100. A route map named `vpn-id-map1` says that any route that matches on routes that are allowed by IP extended community list 1 are subject to two `set` commands. The first `set` command deletes the RT from the route. The second `set` command sets the VPN distinguisher attribute to 111:100.

The `route-map vpn-id-map1 permit 20` command allows other routes, which are not part of the RT-to-VPN distinguisher mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause these routes to be discarded.

Finally, in autonomous system 2000, for the VPNv4 address family, the route map `vpn-id-map1` is applied to routes going out to the neighbor at 192.168.101.1.

Egress ASBR

```
ip extcommunity-list 1 permit rt 101:100
!
route-map vpn-id-map1 permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity vpn-distinguisher 111:100
!
route-map vpn-id-map1 permit 20
!
router bgp 2000
neighbor 192.168.101.1 remote-as 2000
address-family vpnv4
neighbor 192.168.101.1 activate
neighbor 192.168.101.1 route-map vpn-id-map1 out
exit-address-family
!
```

On the ingress ASBR, IP extended community list 51 allows routes with a VPN distinguisher of 111:100. A route map named `vpn-id-rewrite-map1` says that any route that matches on routes that are allowed by IP extended community list 51 are subject to two `set` commands. The first `set` command deletes the VPN distinguisher from the route. The second `set` command sets the RT to 101:1, and that RT is added to the RT list without replacing any RTs.

The **route-map vpn-id-rewrite-map1 permit 20** command allows other routes, which are not part of the VPN distinguisher-to-RT mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause those routes to be discarded.

Finally, in autonomous system 3000, for the VPNv4 address family, the route map named `vpn-id-rewrite-map1` is applied to incoming routes destined for the neighbor at 192.168.0.81.

Ingress ASBR

```
ip extcommunity-list 51 permit vpn-distinguisher 111:100
!
route-map vpn-id-rewrite-map1 permit 10
  match extcommunity 51
  set extcomm-list 51 delete
  set extcommunity rt 101:1 additive
!
route-map vpn-id-rewrite-map1 permit 20
!
router bgp 3000
  neighbor 192.168.0.81 remote-as 3000
  address-family vpnv4
    neighbor 192.168.0.81 activate
    neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in
  exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP-VPN Distinguisher Attribute

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 66: Feature Information for BGP—VPN Distinguisher Attribute

Feature Name	Releases	Feature Information
BGP—VPN Distinguisher Attribute	15.2(1)SY	<p>The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source RTs private from an ASBR in a destination autonomous system. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.</p> <p>The following command was introduced:</p> <ul style="list-style-type: none"> • set extcommunity vpn-distinguisher <p>The following command was modified:</p> <ul style="list-style-type: none"> • show ip bgp vpnv4



BGP-VRF-Aware Conditional Advertisement

The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control to within a virtual routing and forwarding (VRF) instance.

- [Finding Feature Information, page 821](#)
- [Information About BGP VRF-Aware Conditional Advertisement, page 821](#)
- [How to Configure BGP VRF-Aware Conditional Advertisement, page 823](#)
- [Configuration Examples for BGP VRF-Aware Conditional Advertisement, page 825](#)
- [Additional References for BGP VRF-Aware Conditional Advertisement, page 829](#)
- [Feature Information for BGP VRF-Aware Conditional Advertisement, page 830](#)

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 module.

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

Information About BGP VRF-Aware Conditional Advertisement

VRF-Aware Conditional Advertisement

The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control within a virtual routing and forwarding (VRF) instance.

BGP Conditional Advertisement

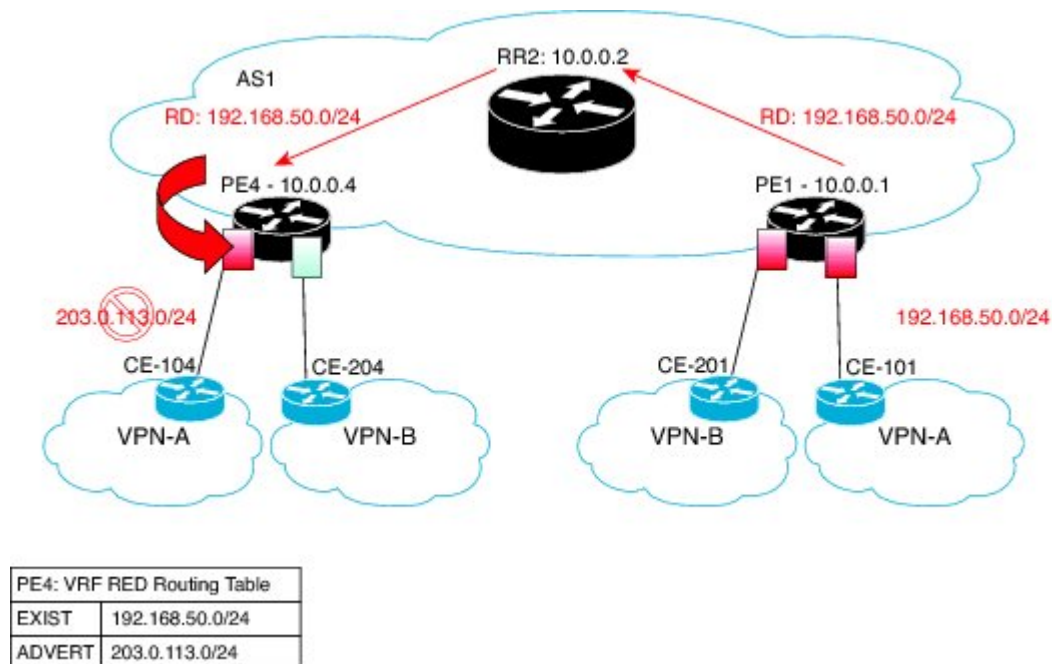
Normally, routes are propagated regardless of the existence of a different route. The BGP conditional advertisement feature uses the **exist-map**, **non-exist-map**, and the **advertise-map** keywords of the **neighbor** command in order to track routes by the route prefix. If a route prefix is not present in output of the **non-exist-map** command, then the route specified by the **advertise-map** is announced. This feature is useful for multihomed networks, in which some prefixes are advertised to one of the providers only if information from the other provider is not present (this indicates a failure in the peering session or partial reachability). The conditional BGP announcements are sent in addition to the normal announcements that a BGP router sends to its peers.

VRF-Aware Conditional Advertisement

This feature extends support for BGP VRF-aware conditional advertisement to the following address families:

- IPv4 unicast
- IPv4 unicast VRF
- IPv6 unicast
- IPv6 unicast VRF

Figure 70: VRF-Based Conditional Advertisement



The figure above shows the IPv4 prefix 192.168.50.0/24 being advertised by a remote CE101 into VRF RED on PE1. The prefix flows as a MP-BGP VPN prefix and is imported into the VRF RED on PE4. On the PE4 the conditions configured by the **exist-map** command relating to this prefix in the BGP VRF RED table becomes the condition to advertise the prefix 203.0.113.0/24 to the CE104, that is, peer-activated under the VRF RED on the PE4. This scenario assumes that 203.0.113.0/24 is in the VRF RED BGP table. If 203.0.113.0/24 is not in the table, this policy is ignored.

- If 192.168.50.0/24 exists in PE4's BGP table, then the 203.0.113.0/24 network is advertised to CE104.
- If 192.168.50.0/24 does not exist in PE4's BGP table, then the 203.0.113.0/24 network is not advertised to CE104.

How to Configure BGP VRF-Aware Conditional Advertisement

Configuring BGP VRF-Aware Conditional Advertisement

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Enter one of the following:
 - **address-family ipv4** [**unicast**] [**vrf vrf-name**]
 - **address-family ipv6** [**unicast**] [**vrf vrf-name**]
5. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *ipv6-address*} **activate**
7. **neighbor** {*ip-address* | *ipv6-address*} **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>Enter one of the following:</p> <ul style="list-style-type: none"> • address-family ipv4 [unicast] [vrf vrf-name] • address-family ipv6 [unicast] [vrf vrf-name] <p>Example:</p> <pre>Device(config-router)# address-family ipv4 vrf VRFRED</pre>	<p>Specifies the IPv4 or IPv6 address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 or IPv6 unicast address family. • The vrf keyword and <i>vrf-name</i> argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 or IPv6 address family configuration mode commands.
Step 5	<p>neighbor {ip-address ipv6-address} remote-as autonomous-system-number</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.0.2.1 remote-as 104</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv4 or IPv6 multiprotocol BGP neighbor table of the local device.</p>
Step 6	<p>neighbor {ip-address ipv6-address} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.0.2.1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 or IPv6 address family with the local device.</p>
Step 7	<p>neighbor {ip-address ipv6-address} advertise-map map-name {exist-map map-name non-exist-map map-name}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 192.0.2.1 advertise-map ADV-1 exist-map EXIST-1</pre>	<p>Enables conditional advertisement towards a neighbor to allow the advertisement of prefixes mapped by the advertise-map command based on the criteria defined under exist or non-exist maps.</p> <ul style="list-style-type: none"> • The advertise-map map-name keyword-argument pair specifies the name of the route map used to define the advertised routes. • The exist-map map-name keyword-argument pair specifies the condition that can be satisfied by a set of routes in the BGP table. If the condition is satisfied then the routes in the BGP table matching those specified in advertise map will be advertised. If the routes matching those specified in exist-map do not exist in the BGP table, those routes will not be advertised. • The non-exist-map map-name keyword-argument pair specifies the condition that is compared to a set of routes in the BGP table. If the routes in the non-exist-map are not present in the BGP table, then the routes matching those specified in advertise map will be advertised. If the routes matching those specified in non-exist-map are present in the BGP table, then the routes matching advertise-map will not be advertised.

	Command or Action	Purpose
Step 8	end Example: Device(config-router-af)# end	Exits address family configuration mode and enters privileged EXEC mode.

What to Do Next

To verify the configuration of the BGP VRF-Aware Conditional Advertisement feature, use the **show bgp ip neighbors** command.

Configuration Examples for BGP VRF-Aware Conditional Advertisement

Example: Configuring BGP VRF-Aware Conditional Advertisement

The following examples use the configuration in figure 1:

CE 101: The source of the prefixes

```
router bgp 101
  bgp log-neighbor-changes
  timers bgp 0 0
  neighbor 172.16.1.2 remote-as 65000
  !
  address-family ipv4
    network 21.21.21.0 mask 255.255.255.0
    network 22.22.22.22 mask 255.255.255.255
    network 31.0.0.0
    network 33.0.0.0
    network 44.0.0.0
    network 192.0.254 mask 255.255.255.0
    network 192.0.2.50
    neighbor 172.16.1.3 activate
  exit-address-family
```

PE 1

```
router bgp 65000
  bgp log-neighbor-changes
  no bgp default ipv4-unicast
  timers bgp 0 0
  neighbor 10.0.0.2 remote-as 65000
  neighbor 10.0.0.2 update-source Loopback0
  !
  address-family ipv4
  exit-address-family
  !
  address-family vpnv4
    neighbor 10.0.0.2 activate
    neighbor 10.0.0.2 send-community both
```

Example: Configuring BGP VRF-Aware Conditional Advertisement

```

exit-address-family
!
address-family ipv4 vrf blue
  neighbor 198.51.100.10 remote-as 201
  neighbor 198.51.100.10 activate
exit-address-family
!
address-family ipv4 vrf red
  neighbor 172.16.1.2 remote-as 101
  neighbor 172.16.1.2 activate
exit-address-family

```

PE 4

```

router bgp 65000
  bgp log-neighbor-changes
  no bgp default ipv4-unicast
  timers bgp 0 0
  neighbor 10.0.0.2 remote-as 65000
  neighbor 10.0.0.2 update-source Loopback0
  !
  address-family ipv4
  exit-address-family
  !
  address-family vpnv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community extended
  exit-address-family
  !
  address-family ipv4 vrf blue
  neighbor 198.51.100.12 remote-as 204
  neighbor 198.51.100.12 activate
  exit-address-family
  !
  address-family ipv4 vrf red
  neighbor 198.51.100.3 remote-as 104
  neighbor 198.51.100.3 activate
  neighbor 198.51.100.3 advertise-map ADV-1 exist-map EXIST-1
  neighbor 198.51.100.3 advertise-map ADV-2 exist-map EXIST-2
  neighbor 198.51.100.3 advertise-map ADV-3 exist-map EXIST-3
  neighbor 198.51.100.3 advertise-map ADV-4 exist-map EXIST-4
  exit-address-family
  !
  ip prefix-list pl-adv-1 seq 5 permit 22.22.22.22/32
  !
  ip prefix-list pl-adv-2 seq 5 permit 44.0.0.0/8 □
  !
  ip prefix-list pl-adv-3 seq 5 permit 33.0.0.0/8
  !
  ip prefix-list pl-adv-4 seq 5 permit 128.16.16.0/24
  !
  ip prefix-list pl-exist-1 seq 5 permit 21.21.21.0/24
  !
  ip prefix-list pl-exist-2 seq 5 permit 41.0.0.0/8 □
  !
  ip prefix-list pl-exist-3 seq 5 permit 31.0.0.0/8
  !
  ip prefix-list pl-exist-4 seq 5 permit 192.168.50.0/24
  !
  route-map EXIST-4 permit 10
  match ip address prefix-list pl-exist-4
  !
  route-map ADV-4 permit 10
  match ip address prefix-list pl-adv-4
  !
  route-map EXIST-2 permit 10
  match ip address prefix-list pl-exist-2
  !
  route-map ADV-2 permit 10
  match ip address prefix-list pl-adv-2
  !
  route-map EXIST-3 permit 10

```

```

match ip address prefix-list pl-exist-3
!
route-map ADV-3 permit 10
match ip address prefix-list pl-adv-3
!
route-map EXIST-1 permit 10
match ip address prefix-list pl-exist-1
!
route-map ADV-1 permit 10
match ip address prefix-list pl-adv-1

```

Example: Verifying BGP VRF-Aware Conditional Advertisement

The following examples use the configuration in figure 1:

CE 101

```
CE101# show ip bgp all
```

```

For address family: IPv4 Unicast
BGP table version is 28, local router ID is 203.0.113.11
  Network          Next Hop          Metric LocPrf Weight Path
 *> 21.21.21.0/24   0.0.0.0           0      0      32768 i
 *> 22.22.22.22/32  0.0.0.0           0      0      32768 i
 *> 31.0.0.0         0.0.0.0           0      0      32768 i
 *> 33.0.0.0         0.0.0.0           0      0      32768 i
 *> 44.0.0.0         0.0.0.0           0      0      32768 i
 *> 192.0.2.254/24  0.0.0.0           0      0      32768 i
 *> 192.0.2.50      0.0.0.0           0      0      32768 i

```

PE 1

```
PE1# show ip bgp all
```

```
For address family: IPv4 Unicast
```

```
For address family: VPNv4 Unicast
```

```

BGP table version is 46, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

```

  Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf red)
 *> 21.21.21.0/24   172.16.1.2        0      0      0 101 i
 *> 22.22.22.22/32  172.16.1.2        0      0      0 101 i
 *> 31.0.0.0         172.16.1.2        0      0      0 101 i
 *> 33.0.0.0         172.16.1.2        0      0      0 101 i
 *> 44.0.0.0         172.16.1.2        0      0      0 101 i
 *> 192.0.2.254/24  172.16.1.2        0      0      0 101 i
 *> 192.0.2.50      172.16.1.2        0      0      0 101 i

```

PE 4

**Note**

The status is Withdraw for the exist-map EXIST-2 because the condition for advertisement has not been met.

```

PE4# show ip bgp all

For address family: VPNv4 Unicast

BGP table version is 82, local router ID is 10.0.0.4

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf red)
*>i 21.21.21.0/24      10.0.0.1          0      100      0 101 i
*>i 22.22.22.22/32    10.0.0.1          0      100      0 101 i
*>i 31.0.0.0          10.0.0.1          0      100      0 101 i
*>i 33.0.0.0          10.0.0.1          0      100      0 101 i
*>i 44.0.0.0          10.0.0.1          0      100      0 101 I   <- missing 41.0.0.0/8

*>i 192.0.2.254/24    10.0.0.1          0      100      0 101 i
*>i 192.0.2.50        10.0.0.1          0      100      0 101 i

PE4# show ip bgp vpnv4 all neighbors 198.51.100.3
...
...
For address family: VPNv4 Unicast
  Translates address family IPv4 Unicast for VRF red
  Session: 198.51.100.3
  BGP table version 48, neighbor version 48/0
  Output queue size : 0
  Index 3, Advertise bit 0
  3 update-group member
  Condition-map EXIST-1, Advertise-map ADV-1, status: Advertise
  Condition-map EXIST-2, Advertise-map ADV-2, status: Withdraw
  Condition-map EXIST-3, Advertise-map ADV-3, status: Advertise
  Condition-map EXIST-4, Advertise-map ADV-4, status: Advertise
  Slow-peer detection is disabled
  ...
...
PE4#

PE4# show ip bgp vpnv4 all update-group

...
...
BGP version 4 update-group 3, external, Address Family: VPNv4 Unicast
  BGP Update version : 48/0, messages 0
  Condition-map EXIST-1, Advertise-map ADV-1, status: Advertise
  Condition-map EXIST-2, Advertise-map ADV-2, status: Withdraw
  Condition-map EXIST-3, Advertise-map ADV-3, status: Advertise
  Condition-map EXIST-4, Advertise-map ADV-4, status: Advertise
  Topology: red, highest version: 47, tail marker: 47
  Format state: Current working (OK, last not in list)
                 Refresh blocked (not in list, last not in list)
  Update messages formatted 4, replicated 4, current 0, refresh 0, limit 1000
  Number of NLRIs in the update sent: max 3, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 1 member:
    198.51.100.3

```

CE 104**Note**

Prefix 44.0.0.0 is missing as 41.0.0.0/8 does not appear in PE 4 to trigger the advertisement to CE 104. The state is Withdraw.

```
CE104# show ip bgp all
```

```
For address family: IPv4 Unicast
```

```
BGP table version is 45, local router ID is 198.51.100.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```

      Network          Next Hop           Metric LocPrf Weight Path
* >  21.21.21.0/24     104.0.0.1          0       65000 101    i
* >  22.22.22.22/32   104.0.0.1          0       65000 101    i
* >  31.0.0.0          104.0.0.1          0       65000 101    i
* >  33.0.0.0          104.0.0.1          0       65000 101    i
* >  192.0.2.254/24   104.0.0.1          0       65000 101    i
* >  192.0.2.50       104.0.0.1          0       65000 101    i

```

Additional References for BGP VRF-Aware Conditional Advertisement

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/support</p>

Feature Information for BGP VRF-Aware Conditional Advertisement

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 67: Feature Information for BGP VRF-Aware Conditional Advertisement

Feature Name	Releases	Feature Information
BGP VRF-Aware Conditional Advertisement	15.2(1)SY	The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control to within a virtual routing and forwarding (VRF) instance.



BGP Diverse Path Using a Diverse-Path Route Reflector

The BGP Diverse Path Using a Diverse-Path Route Reflector feature allows Border Gateway Protocol (BGP) to distribute an alternative path other than the best path between BGP speakers when route reflectors are deployed. This feature is meant to provide path diversity within an autonomous system (AS), within a single cluster only. That is, a route reflector is allowed to advertise the diverse path to its client peers only.

- [Finding Feature Information, page 831](#)
- [Prerequisites for BGP Diverse Path Using a Diverse-Path Route Reflector, page 832](#)
- [Restrictions for BGP Diverse Path Using a Diverse-Path Route Reflector, page 832](#)
- [Information About BGP Diverse Path Using a Diverse-Path Reflector, page 832](#)
- [How to Configure a BGP Diverse-Path Route Reflector, page 835](#)
- [Configuration Examples for BGP Diverse Path Using a Diverse-Path Route Reflector, page 839](#)
- [Additional References, page 841](#)
- [Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector, page 842](#)

Finding Feature Information

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

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

Prerequisites for BGP Diverse Path Using a Diverse-Path Route Reflector

You should understand the BGP Best External feature.

Restrictions for BGP Diverse Path Using a Diverse-Path Route Reflector

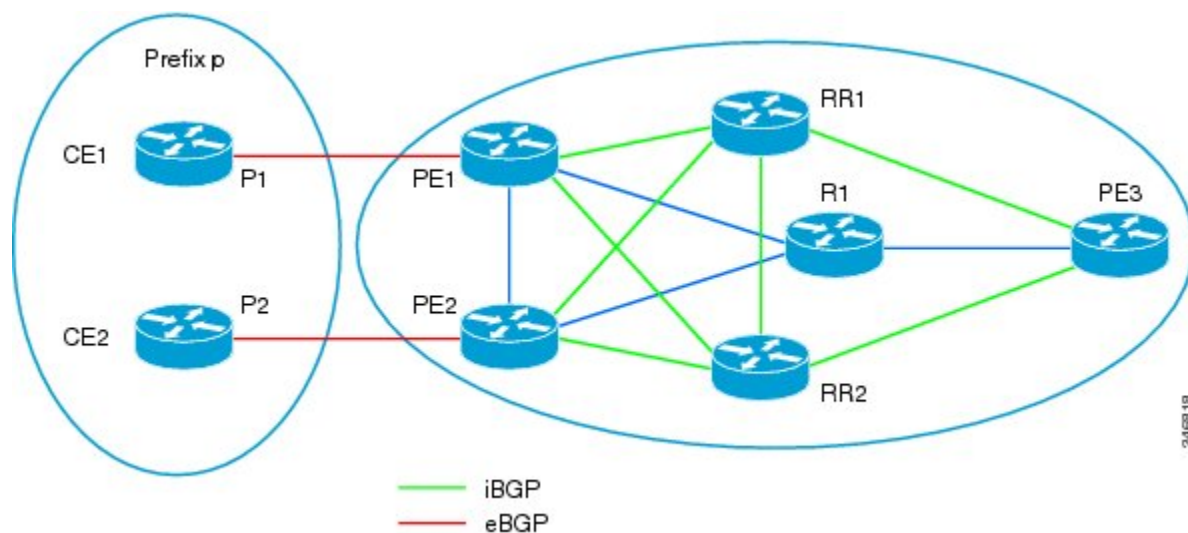
- A diverse path can be configured on a route reflector only.
- Only one shadow route reflector is allowed per existing route reflector, which will calculate one additional best path (the second best path). That is, only one additional plane (topology) is configured.
- Path diversity is configured within an AS, within a single route reflector cluster. That is, the route reflector will advertise the diverse path to its route reflector client peers only.
- Diverse path functionality is not supported on a route server.

Information About BGP Diverse Path Using a Diverse-Path Route Reflector

Limitation that a BGP Diverse Path Overcomes

As a path vector routing protocol, BGP-4 requires a router to advertise to its neighbors only the best path for a destination. However, multiple paths to the same destination would allow mechanisms that can improve resilience, quickly recover from failures, and load balance, for example.

The use of route reflectors is one of the main reasons for poor path diversity within an autonomous system (AS). In a network with route reflectors, even if a prefix is learned from multiple egress routers, the route reflector reflects only the best path to its clients. The figure below shows how deploying route reflectors might reduce path diversity in an AS, even when the BGP Best External feature is deployed.



In the figure above, P1 and P2 are diverse paths for prefix p. Assume Path 2 (P2) has a lower MED and higher local preference than P1. The BGP Best External feature on PE1 will make sure that P1 is propagated to the route reflectors, regardless of P2 having a lower MED and higher local preference. The route reflectors will have path diversity; they will learn both P1 and P2 with different exit points PE1 and PE2 (assuming that PE1 and PE2 have the `set ip next-hop self` command configured). However, both route reflectors select the best path as P2 due to its lower MED/higher local preference and advertise it to PE3. PE3 will not learn P1 (that is, PE3 will not learn about existing path diversity).

The BGP Diverse Path Using a Diverse-Path Route Reflector feature is a way to resolve that limitation and achieve path diversity.

BGP Diverse Path Using a Diverse-Path Route Reflector

The BGP Diverse Path Using a Diverse-Path Route Reflector feature overcomes the lack of path diversity in an AS containing route reflectors. This feature is meant to provide path diversity within an AS, within a single cluster only. That is, a route reflector is allowed to advertise the diverse path to its client peers only.

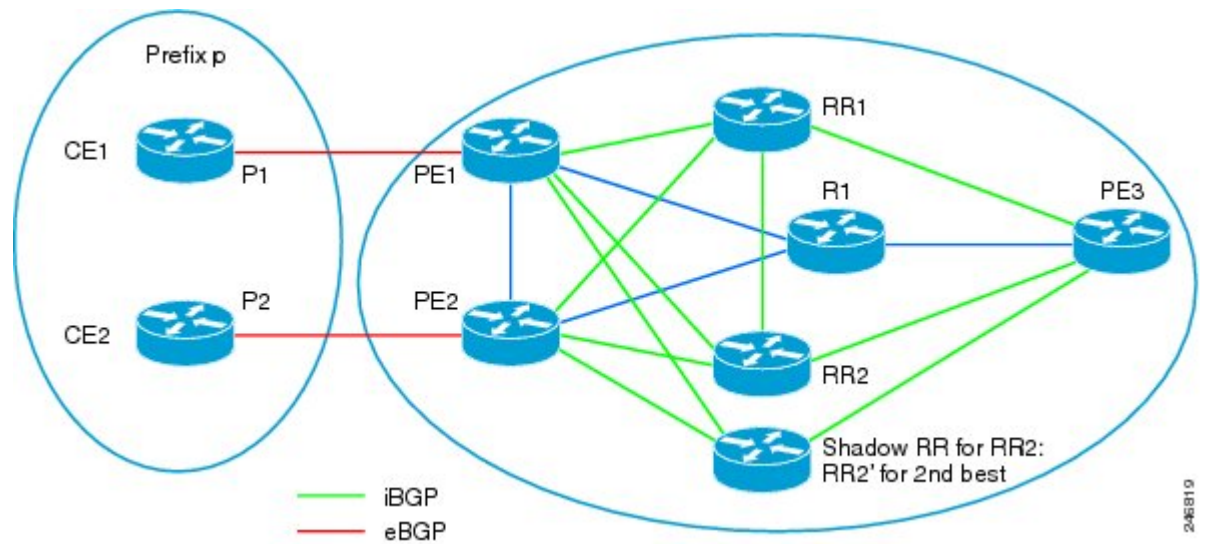
For each route reflector in the AS, a *shadow route reflector* is added to distribute the *second best path*, also known as the *diverse path*. The figure below shows the shadow route reflector for RR2. The shadow route reflector improves path diversity because PE3 can now learn both P1 (from RR1/RR2) and learn P2 from the shadow route reflector.



Note

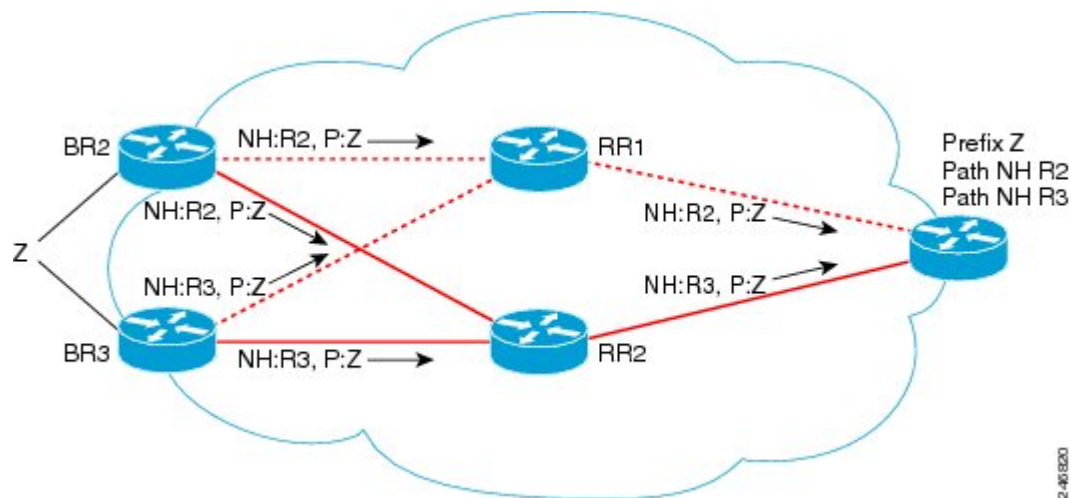
The primary route reflector and shadow route reflector must have the exact same connections (physical/control plane) to the rest of the routers in the network.

Shadow route reflectors can be both control plane route reflectors and data plane route reflectors.



The figure below shows a diverse path in greater detail, indicating the next hops:

- BR2 announces to RR1 and shadow RR2 that R2 (BR2) is the Next Hop for those who want to reach Prefix Z. Likewise, BR3 announces to RR1 and shadow RR2 that R3 (BR3) is the Next Hop for those who want to reach Prefix Z
- RR1 sends a packet to BR1 announcing that the Next Hop is R2 if BR1 wants to reach Prefix Z. The second best path (or diverse path) comes from shadow RR2, which sends a packet to BR1 announcing that the Next Hop is R3 if BR1 want to reach Prefix Z.
- At BR1 (far right), we see there are two (diverse) paths to Prefix Z.



Triggers to Compute a BGP Diverse Path

Computation of a diverse path per address family is triggered by any of the following commands:

- **bgp additional-paths install**

- **bgp additional-paths select**
- **maximum-paths ebgp**
- **maximum-paths ibgp**

The **bgp additional-paths install** command will install the type of path that is specified in the **bgp additional-paths select** command. If the **bgp additional-paths select** command specifies both keyword options (**best-external** and **backup**), the system will install a backup path.

The **maximum-paths ebgp** and **maximum-paths ibgp** commands trigger a multipath computation, and multipaths are automatically installed as primary paths.

On the other hand, the **bgp additional-paths install** command triggers computation of a backup path or best-external path.

If the **bgp additional-paths select** command is not configured, the **bgp additional-paths install** command will trigger both computation and installation of a backup path (as is done with the BGP PIC feature).

IGP Metric Check

Disabling the Interior Gateway Protocol (IGP) metric check and configuring the BGP Diverse Path feature are independent of each other. One does not imply the other. That is, configuring **bgp bestpath igp-metric ignore** does not imply that the BGP Diverse Path feature is enabled. Conversely, enabling the BGP Diverse Path feature might not require that **bgp bestpath igp-metric ignore** be configured (because, for example, the route reflector and shadow route reflector are co-located).

The **bgp bestpath igp-metric ignore** command can be configured at route reflectors and provider edges (PEs).

**Note**

Per-VRF functionality for the **bgp bestpath igp-metric ignore** command is not supported. If you use it anyway, it is at your own risk.

Route Reflector Determination

If a router's configuration includes either one of the following commands, the router is a route reflector:

- **bgp cluster-id**
- **neighbor route-reflector-client**

How to Configure a BGP Diverse-Path Route Reflector

Determining Whether You Need to Disable the IGP Metric Check

Before you configure a shadow route reflector in order to get a BGP diverse path, determine whether you need to disable the IGP metric check. The IGP metric is a configurable value indicating physical distance, and is used by an Interior Gateway Protocol, such as Open Shortest Path First (OSPF), Enhanced Interior

Gateway Routing Protocol (EIGRP), or Routing Information Protocol (RIP). A smaller IGP metric is preferred over a larger IGP metric.

The locations of the route reflector and shadow route reflector determine whether or not you need to disable the IGP metric check, as follows:

- When the route reflector and shadow route reflector are colocated—They have the same IP subnetwork address and are connected to the Ethernet switch with different links. Failure of such a link is equivalent to the route reflector going down. When RRs are colocated, their IGP metrics cannot be different from each other; and therefore there is no need to disable the IGP metric check during the best path calculation at any route reflector. Because there is no need to disable the IGP metric check, the first plane route reflectors do not need to be upgraded to Cisco IOS XE Release 3.4S.
- When the shadow route reflector is in a different IGP place from the route reflector (it is not colocated with its best path route reflector)--In this case, the IGP metric check is ignored on both the best path route reflector and shadow route reflector when the best path and second best path are being calculated. The IGP metric check must be disabled on the primary route reflector by configuring the **bgp bestpath igp-metric ignore** command. This command is available beginning with Cisco IOS XE Release 3.4S, which means you need to upgrade to that release.

Configuring the Route Reflector for BGP Diverse Path

Perform this task to configure a route reflector for the BGP Diverse Path feature. This task specifies the IPv4 address family, but other address families are also supported.

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**
6. **neighbor** *ip-address* **activate**
7. **maximum-paths ibgp** *number-of-paths*
8. **bgp bestpath igp-metric ignore**
9. **bgp additional-paths select** [**backup**]
10. **bgp additional-paths install**
11. **neighbor** *ip-address* **route-reflector-client**
12. **neighbor** *ip-address* **advertise diverse-path** [**backup**] [**mpath**]
13. **end**
14. **show ip bgp neighbor** *ip-address* **advertised-routes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 1</pre>	Enters router configuration mode for the BGP routing process.
Step 4	<p>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 10.1.1.1 remote-as 1</pre>	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	<p>address-family ipv4 unicast</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast</pre>	<p>Specifies the address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> • Supported address families are IPv4 unicast, VPNv4 unicast, IPv6 unicast, VPNv6 unicast, IPv4+label, and IPv6+label.
Step 6	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.1.1.1 activate</pre>	Enables the exchange of information with a BGP neighbor.
Step 7	<p>maximum-paths ibgp <i>number-of-paths</i></p> <p>Example:</p> <pre>Device(config-router-af)# maximum-paths ibgp 4</pre>	Controls the maximum number of parallel Internal BGP (IBGP) routes that can be installed in a routing table.
Step 8	<p>bgp bestpath igp-metric ignore</p> <p>Example:</p> <pre>Device(config-router-af)# bgp bestpath igp-metric ignore</pre>	Configures the system to ignore the Interior Gateway Protocol (IGP) metric during BGP best path selection.

	Command or Action	Purpose
Step 9	bgp additional-paths select [backup] Example: <pre>Device(config-router-af)# bgp additional-paths select backup</pre>	Configures the system to calculate a second BGP best path.
Step 10	bgp additional-paths install Example: <pre>Device(config-router-af)# bgp additional-paths install</pre>	Enables BGP to calculate a backup path for a given address family and to install it into the routing information base (RIB) and Cisco Express Forwarding (CEF).
Step 11	neighbor ip-address route-reflector-client Example: <pre>Device(config-router-af)# neighbor 10.1.1.1 route-reflector-client</pre>	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 12	neighbor ip-address advertise diverse-path [backup] [mpath] Example: <pre>Device(config-router-af)# neighbor 10.1.1.1 advertise diverse-path backup</pre>	(Optional) Configures a neighbor to receive the diverse path in an advertisement.
Step 13	end Example: <pre>Device(config-router-af)# end</pre>	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.
Step 14	show ip bgp neighbor ip-address advertised-routes Example: <pre>Device# show ip bgp neighbor 10.1.1.1 advertised-routes</pre>	(Optional) Displays the routes advertised to the specified neighbor.

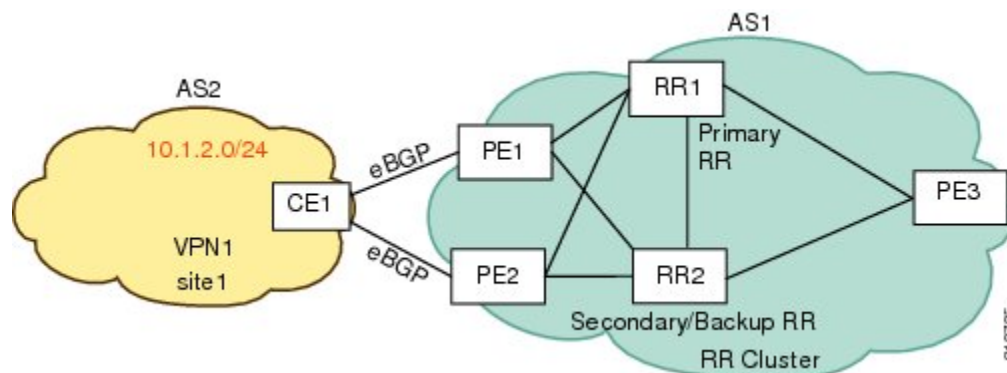
Configuration Examples for BGP Diverse Path Using a Diverse-Path Route Reflector

Example: Configuring BGP Diverse Path Where Additional Path Is the Backup Path

Diverse path functionality is contained within a single cluster; that is, only the clients of a route reflector can be configured to advertise the diverse path. A diverse path is advertised to the clients of a route reflector only if the client is configured to get the additional path.

A shadow route reflector can be added to calculate and advertise the additional path, or an existing route reflector can be configured to calculate and advertise the additional path. In the figure below, instead of adding a shadow route reflector, RR2 (the existing backup RR) is configured to calculate the additional path and advertise it to a particular neighbor.

In the figure below, assume that from the route reflectors, the path to CE1 via PE1 is preferred over the path via PE2. Without the diverse path feature, both route reflectors will advertise to PE3 that the path to CE1 is via PE1. If the connection between RR1 and PE1 fails (or the path between PE1 and CE1 fails), there is no other path.



In the following configuration example based on the figure above, RR2 is configured with an additional path, which is a backup path.

If RR1 and RR2 are not colocated, you must configure the **bgp bestpath igp-metric ignore** command before the additional path is calculated. (If RR1 and RR2 are colocated, do not configure that command.)

The **bgp additional-paths select backup** command triggers calculation of the backup path at RR2, which is the path via PE2.

The **bgp additional-paths install** command installs the backup path if RR2 is in the forwarding plane. (Do not configure this command if RR2 is in the control plane.)

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup** command on RR2. This command triggers advertisement of the backup path to PE3. PE3 will learn the best path, (which is the path via PE1) from RR1, and it will learn the backup path from RR2.

RR2

```

router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp bestpath igp-metric ignore
 bgp additional-paths select backup
 bgp additional-paths install
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup

```

Example: Configuring BGP Diverse Path Where Additional Path Is the Multipath

In the following example based on the figure above, assume that paths toward CE1 via PE1 and PE2 are multipaths. The **maximum-paths ibgp** command will trigger calculation of multipaths.

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path mpath** command on RR2. This command will trigger advertisement of the multipath, that is, the second best path, to PE3. PE3 will learn the best path, path via PE1 from RR1, and will learn second best path from RR2.

RR2

```

router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path mpath

```

Example: Configuring BGP Diverse Path Where Both Multipath and Backup Path Calculations Are Triggered

The following example is based on the figure above. The **maximum-paths ibgp** command will trigger calculation of multipaths. When both multipath and backup path calculations are triggered, the backup path and the second multipath (which is the second best path) are the same paths and it will be installed as the active path, regardless of whether the route reflector is in the control plane or forwarding plane.

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup mpath** command on RR2. This command causes RR2 to advertise the second best path, which is the second multipath, to PE3.

RR2

```

router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp additional-paths select backup
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup mpath

```

Example: Configuring Triggering Computation and Installation of a Backup Path

When the **bgp additional-paths install** command is configured without configuring **bgp additional-paths select backup**, the former command will trigger both computation and installation of the backup path (as it is with the existing BGP PIC feature).

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup** command on RR2. This command will trigger advertisement of a backup path to PE3. PE3 will learn the best path, a path via PE1 from RR1, and it will learn a backup path from RR2.

RR2

```
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp additional-paths install
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Configuring BGP Best External Path on a Route Reflector for Intercluster	BGP Best External module
BGP configuration tasks	Cisco IOS XE IP Routing: BGP Configuration Guide

Standards

Standard	Title
draft-ietf-grow-diverse-bgp-path-dist-02.txt	<i>Distribution of Diverse BGP Paths</i>

MIBs

MIB	MIBs Link
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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4271	A Border Gateway Protocol 4 (BGP-4)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector

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 [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 68: Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector

Feature Name	Releases	Feature Information
BGP Diverse Path Using a Diverse-Path Route Reflector	15.2(1)SY	<p>This feature allows BGP to distribute an alternative path other than the best path between BGP speakers when route reflectors are deployed.</p> <p>The following commands were introduced:</p> <ul style="list-style-type: none">• bgp additional-paths select• bgp bestpath igp-metric ignore• debug ip bgp igp-metric ignore• neighbor advertise best-external• neighbor advertise diverse-path



BGP Graceful Shutdown

The BGP Graceful Shutdown feature reduces or eliminates the loss of traffic along a link being shut down for maintenance. Routers always have a valid route available during the convergence process. This feature is used primarily for maintenance on a link between a Provider Edge (PE), PE-PE, PE- Route Reflector (RR), PE-Customer Edge (CE) and CE.

- [Finding Feature Information, page 845](#)
- [Information About BGP Graceful Shutdown, page 845](#)
- [How to Configure BGP Graceful Shutdown, page 846](#)
- [Configuration Examples for BGP Graceful Shutdown, page 852](#)
- [Additional References, page 854](#)
- [Feature Information for BGP Graceful Shutdown, page 855](#)

Finding Feature Information

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

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

Information About BGP Graceful Shutdown

Purpose and Benefits of BGP Graceful Shutdown

There are times when planned maintenance operations cause routing changes in BGP. After the shutdown of eBGP and iBGP peering sessions between autonomous system border routers (ASBRs), BGP devices are temporarily unreachable during BGP convergence. The goal of gracefully shutting down one or more BGP sessions is to minimize traffic loss during the planned shutdown and subsequent reestablishment of the sessions.

The BGP Graceful Shutdown feature reduces or eliminates the loss of inbound or outbound traffic flows that were initially forwarded along the peering link that is being shut down for maintenance. This feature is primarily for PE-CE, PE-RR and PE-PE links. Lowering the local preference for paths received over the session being shutdown renders the affected paths less preferred by the BGP decision process, but still allows the paths to be used during the convergence while alternative paths are propagated to the affected devices. Therefore, devices always have a valid route available during the convergence process.

The feature also allows vendors to provide a graceful shutdown mechanism that does not require any router reconfiguration at maintenance time. The benefits of the BGP Graceful Shutdown feature are fewer lost packets and less time spent reconfiguring devices.

GSHUT Community

The GSHUT community is a well-known community used in conjunction with the BGP Graceful Shutdown feature. The GSHUT community attribute is applied to a neighbor specified by the **neighbor shutdown graceful** command, thereby gracefully shutting down the link in an expected number of seconds. The GSHUT community is always sent by the GSHUT initiator.

The GSHUT community is specified in a community list, which is referenced by a route map and then used to make policy routing decisions.

The GSHUT community can also be used in the **show ip bgp community** command to limit output to GSHUT routes.

BGP GSHUT Enhancement

The BGP Graceful Shutdown (GSHUT) Enhancement feature enables graceful shutdown of either all neighbors or only virtual routing and forwarding (VRF) neighbors across BGP sessions. To enable the BGP GSHUT enhancement feature on the device, you must configure either the **community** keyword or the **local-preference** keyword in the **bgp graceful-shutdown all** command. Use the **activate** keyword to activate graceful shutdown either across all neighbors or only across all VRF neighbors, across all BGP sessions.

How to Configure BGP Graceful Shutdown

Shutting Down a BGP Link Gracefully

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number*
5. **neighbor** {*ipv4-address* | *ipv6-address* | *peer-group-name*} **shutdown graceful** *seconds* {**community** *value* [**local-preference** *value*] | **local-preference** *value*}
6. **end**
7. **show ip bgp community gshut**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 5000	Configures a BGP routing process.
Step 4	neighbor { <i>ipv4-address</i> <i>ipv6-address</i> } remote-as <i>number</i> Example: Device(config-router)# neighbor 2001:db8:3::1 remote-as 5500	Configures the autonomous system (AS) to which the neighbor belongs.
Step 5	neighbor { <i>ipv4-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } shutdown graceful <i>seconds</i> { community <i>value</i> [local-preference <i>value</i>] local-preference <i>value</i> } Example: Device(config-router)# neighbor 2001:db8:3::1 shutdown graceful 600 community 1200 local-preference 300	Configures the device to gracefully shut down the link to the specified peer in the specified number of seconds; advertises the route with the GSHUT (Graceful Shutdown) community; and advertises the route with another community or specifies a local preference value for the route, or both. <ul style="list-style-type: none"> • Make sure to specify an adequate amount of time for iBGP peers to converge and to choose an alternate path as the best path. • If the graceful keyword is used in the neighbor shutdown command, at least one of the two attributes (a community or local preference) must be configured. You may configure both attributes. • If the graceful keyword is used in the neighbor shutdown command, the route is advertised with the GSHUT community by default. You may also set one other community for policy routing purposes. • In this particular example, the route to the neighbor is configured to shut down in 600 seconds, is advertised with the GSHUT community and community 1200, and is configured with a local preference of 300. • The device receiving the advertisement looks at the community value(s) of the route and optionally uses the community value to

	Command or Action	Purpose
		<p>apply routing policy. Filtering routes based on a community is done with the ip community-list command and a route map.</p> <ul style="list-style-type: none"> • During the graceful shutdown, the neighbor shutdown command is not nvgened. After the timer expires, SHUTDOWN is nvgened.
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Returns to EXEC mode.
Step 7	<p>show ip bgp community gshut</p> <p>Example:</p> <pre>Device# show ip bgp community gshut</pre>	(Optional) Displays information about the routes that are advertised with the well-known GSHUT community.

Filtering BGP Routes Based on the GSHUT Community

Perform this task on a BGP peer to the device where you enabled the BGP Graceful Shutdown feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number*
5. **neighbor** {*ipv4-address* | *ipv6-address*} **activate**
6. **neighbor** {*ipv4-address* | *ipv6-address*} **send-community**
7. **exit**
8. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
9. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
10. **exit**
11. **ip community-list** {*standard* | *standard list-name*} {**deny** | **permit**} **gshut**
12. **router bgp** *autonomous-system-number*
13. **neighbor** *address* **route-map** *map-name* **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 2000	Configures a BGP routing process.
Step 4	neighbor { <i>ipv4-address</i> <i>ipv6-address</i> } remote-as <i>number</i> Example: Device(config-router)# neighbor 2001:db8:4::1 remote-as 1000	Configures the autonomous system (AS) to which the neighbor belongs.
Step 5	neighbor { <i>ipv4-address</i> <i>ipv6-address</i> } activate Example: Device(config-router)# neighbor 2001:db8:4::1 activate	Activates the neighbor.
Step 6	neighbor { <i>ipv4-address</i> <i>ipv6-address</i> } send-community Example: Device(config-router)# neighbor 2001:db8:4::1 send-community	Enables BGP community exchange with the neighbor.
Step 7	exit Example: Device(config-router)# exit	Exits router configuration mode.
Step 8	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# route-map RM_GSHUT deny 10	Configures a route map to permit or deny routes for policy routing.

	Command or Action	Purpose
Step 9	<p>match community {<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i> [exact]}</p> <p>Example:</p> <pre>Device(config-route-map)# match community GSHUT</pre>	Configures that the routes that match ip community-list GSHUT will be policy routed.
Step 10	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route-map configuration mode.
Step 11	<p>ip community-list {<i>standard</i> standard <i>list-name</i>} {deny permit} gshut</p> <p>Example:</p> <pre>Device(config)# ip community-list standard GSHUT permit gshut</pre>	<p>Configures a community list and permits or denies routes that have the GSHUT community to the community list.</p> <ul style="list-style-type: none"> • If you specify other communities in the same statement, there is a logical AND operation and all communities in the statement must match the communities for the route in order for the statement to be processed.
Step 12	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 2000</pre>	Configures a BGP routing process.
Step 13	<p>neighbor <i>address</i> route-map <i>map-name</i> in</p> <p>Example:</p> <pre>Device(config)# neighbor 2001:db8:4::1 route-map RM_GSHUT in</pre>	<p>Applies the route map to incoming routes from the specified neighbor.</p> <ul style="list-style-type: none"> • In this example, the route map named RM_GSHUT denies routes from the specified neighbor that have the GSHUT community.

Configuring BGP GSHUT Enhancement

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-shutdown all** {**neighbors** | **vrfs**} *shutdown-time* {**community** *community-value* [**local-preference** *local-pref-value*] | **local-preference** *local-pref-value* [**community** *community-value*]}
5. **bgp graceful-shutdown all** {**neighbors** | **vrfs**} **activate**
6. **end**
7. **show ip bgp**
8. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode to create or configure a BGP routing process.
Step 4	bgp graceful-shutdown all { neighbors vrfs } <i>shutdown-time</i> { community <i>community-value</i> [local-preference <i>local-pref-value</i>] local-preference <i>local-pref-value</i> [community <i>community-value</i>]} Example: Device(config-router)# bgp graceful-shutdown all neighbors 180 local-preference 20 community 10	Enables the BGP GSHUT enhancement feature on the device.

	Command or Action	Purpose
Step 5	bgp graceful-shutdown all {neighbors vrfs} activate Example: <pre>Device(config-router)# bgp graceful-shutdown all neighbors activate</pre>	Activates graceful shutdown across all neighbors or only across VRF neighbors for BGP sessions.
Step 6	end Example: <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 7	show ip bgp Example: <pre>Device# show ip bgp neighbors 10.2.2.2 include shutdown</pre>	Displays entries in the BGP routing table.
Step 8	show running-config Example: <pre>Device# show running-config session router bgp</pre>	Displays running configuration on the device.

Configuration Examples for BGP Graceful Shutdown

Example: Shutting Down a BGP Link Gracefully

Graceful Shutdown While Setting a Local-Preference

This example gracefully shuts down the link to the specified neighbor in 600 seconds, adds the GSHUT community to the route, and sets a local preference of 500 for the route.

```
router bgp 1000
neighbor 2001:db8:5::1 remote-as 2000
neighbor 2001:db8:5::1 shutdown graceful 600 local-preference 500
neighbor 2001:db8:5::1 send-community
exit
```

Graceful Shutdown While Setting an Additional Community

This example gracefully shuts down the link to the specified neighbor in 600 seconds, and adds the GSHUT community and numbered community to the route.

```
router bgp 1000
```

```
neighbor 2001:db8:5::1 remote-as 2000
neighbor 2001:db8:5::1 shutdown graceful 600 community 1400
neighbor 2001:db8:5::1 send-community
exit
```

Graceful Shutdown while Setting an Additional Community and Local-Preference

This example gracefully shuts down the link to the specified neighbor in 600 seconds, adds the GSHUT community and the numbered community to the route, and sets a local preference of 500 to the route.

```
router bgp 1000
neighbor 2001:db8:5::1 remote-as 2000
neighbor 2001:db8:5::1 shutdown graceful 600 community 1400 local-preference 500
neighbor 2001:db8:5::1 send-community
exit
```

Example: Filtering BGP Routes Based on the GSHUT Community

In addition to being able to gracefully shut down a BGP route, another use of the GSHUT community is to configure a community list to filter routes with this community from getting into the BGP routing table.

This example illustrates how to use a community list to filter incoming BGP routes based on the GSHUT community. In this example, a route map named RM_GSHUT denies routes based on a standard community list named GSHUT. The community list contains routes with the GSHUT community. The route map is then applied to incoming routes from the neighbor at 2001:db8:4::1.

```
router bgp 2000
neighbor 2001:db8:4::1 remote-as 1000
neighbor 2001:db8:4::1 activate
neighbor 2001:db8:4::1 send-community
exit
route-map RM_GSHUT deny 10
match community GSHUT
exit
ip community-list standard GSHUT permit gshut
router bgp 2000
neighbor 2001:db8:4::1 route-map RM_GSHUT in
```

Example: BGP GSHUT Enhancement

The following example shows how to enable and activate the BGP GSHUT enhancement feature across all neighbors. In this example, the neighbors are configured to gracefully shutdown within the specified duration of 180 seconds.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# bgp graceful-shutdown all neighbors 180 local-preference 20 community 10
Device(config-router)# bgp graceful-shutdown all neighbors activate
Device(config-router)# end
```

Following is sample output from the **show ip bgp** command, which displays the graceful shutdown time for each neighbor. In this example, there are two IPv4 neighbors configured with IP address 10.2.2.2 and 172.16.2.1 and one VRF neighbor, tagged v1, is configured with IP address 192.168.1.1.

```
Device# show ip bgp neighbors 10.2.2.2 | include shutdown
```

```
Graceful Shutdown Timer running, schedule to reset the peer in 00:02:47 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10
```

```
Device# show ip bgp neighbors 172.16.2.1 | include shutdown
```

```
Graceful Shutdown Timer running, schedule to reset the peer in 00:02:38 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10
```

```
Device# show ip bgp vpnv4 vrf v1 neighbors 192.168.1.1 | include shutdown
```

```
Graceful Shutdown Timer running, schedule to reset the peer in 00:01:45 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10
```

Following is sample output from the **show running-config** command, which displays information associated with the BGP session in router configuration mode:

```
Device# show running-config | session router bgp
```

```
router bgp 65000
bgp log-neighbor-changes
bgp graceful-shutdown all neighbors 180 local-preference 20 community 10
network 10.1.1.0 mask 255.255.255.0
neighbor 10.2.2.2 remote-as 40
neighbor 10.2.2.2 shutdown
neighbor 172.16.2.1 remote-as 10
neighbor 172.16.2.1 shutdown
!
address-family vpnv4
neighbor 172.16.2.1 activate
neighbor 172.16.2.1 send-community both
exit-address-family
!
address-family ipv4 vrf v1
neighbor 192.168.1.1 remote-as 30
neighbor 192.168.1.1 shutdown
neighbor 192.168.1.1 activate
neighbor 192.168.1.1 send-community both
exit-address-family
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6198	<i>Requirements for the Graceful Shutdown of BGP Sessions</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Graceful Shutdown

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 [http://www.cisco.com/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 69: Feature Information for BGP Graceful Shutdown

Feature Name	Releases	Feature Information
BGP Graceful Shutdown	15.2(1)SY	The BGP Graceful Shutdown feature reduces or eliminates the loss of traffic along a link being shut down for maintenance. Routers always have a valid route available during the convergence process. The following commands were modified: ip community-list , neighbor shutdown , show ip bgp community , and show ip bgp vpv4 .
BGP GSHUT Enhancement	15.2(1)SY	The BGP Graceful Shutdown (GSHUT) Enhancement feature enables graceful shutdown of either all neighbors or only virtual routing and forwarding (VRF) neighbors across BGP sessions. The following command was introduced: bgp graceful-shutdown all .

