



IP Routing: BGP Configuration Guide, Cisco IOS Release 12.2SR

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 NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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

Cisco and the Cisco logo are trademarks 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: 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)

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.

© 2012 Cisco Systems, Inc. All rights reserved.



CONTENTS

Cisco BGP Overview	1
Finding Feature Information	1
Prerequisites for Cisco BGP	1
Restrictions for Cisco BGP	1
Information About Cisco BGP	2
BGP Version 4 Functional Overview	2
BGP Autonomous Systems	3
BGP Autonomous System Number Formats	4
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	12
VPNv4 Address Family	13
L2VPN Address Family	13
BGP CLI Removal Considerations	14
Where to Go Next	15
Additional References	15
Feature Information for Cisco BGP Overview	17
Configuring a Basic BGP Network	23
Finding Feature Information	23
Prerequisites for Configuring a Basic BGP Network	23
Restrictions for Configuring a Basic BGP Network	23
Information About Configuring a Basic BGP Network	24
BGP Version 4	24

BGP Router ID	25
BGP-Speaker and Peer Relationships	25
BGP Autonomous System Number Formats	25
Cisco Implementation of 4-Byte Autonomous System Numbers	28
BGP Peer Session Establishment	29
Cisco Implementation of BGP Global and Address Family Configuration Commands	29
BGP Session Reset	31
BGP Route Aggregation	31
BGP Aggregation Route AS-SET Information Generation	32
Routing Policy Change Management	32
Conditional BGP Route Injection	33
BGP Peer Groups	34
BGP Backdoor Routes	34
Peer Groups and BGP Update Messages	35
BGP Update Group	35
BGP Dynamic Update Group Configuration	35
BGP Peer Templates	35
Inheritance in Peer Templates	36
Peer Session Templates	37
Peer Policy Templates	38
BGP IPv6 Neighbor Activation Under the IPv4 Address Family	39
How to Configure a Basic BGP Network	39
Configuring a BGP Routing Process	40
Troubleshooting Tips	43
Configuring a BGP Peer	43
Troubleshooting Tips	46
What to Do Next	46
Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers	46
Troubleshooting Tips	50
Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers	50
Configuring a BGP Peer for the IPv4 VRF Address Family	53
Troubleshooting Tips	57
Customizing a BGP Peer	57
Removing BGP Configuration Commands Using a Redistribution	62

Monitoring and Maintaining Basic BGP	64
Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing	64
Resetting and Displaying Basic BGP Information	67
Aggregating Route Prefixes Using BGP	69
Redistributing a Static Aggregate Route into BGP	69
Configuring Conditional Aggregate Routes Using BGP	71
Suppressing and Unsuppressing Advertising Aggregated Routes Using BGP	72
Suppressing Inactive Route Advertisement Using BGP	74
Conditionally Advertising BGP Routes	76
Originating BGP Routes	79
Advertising a Default Route Using BGP	79
Troubleshooting Tips	81
Conditionally Injecting BGP Routes	81
Troubleshooting Tips	85
Originating BGP Routes Using Backdoor Routes	85
Configuring a BGP Peer Group	87
Configuring Peer Session Templates	89
Configuring a Basic Peer Session Template	89
What to Do Next	91
Configuring Peer Session Template Inheritance with the inherit peer-session Command	92
What to Do Next	94
Configuring Peer Session Template Inheritance with the neighbor inherit peer-session Command	94
What to Do Next	95
Configuring Peer Policy Templates	95
Configuring Basic Peer Policy Templates	96
What to Do Next	97
Configuring Peer Policy Template Inheritance with the inherit peer-policy Command	98
Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command	100
Monitoring and Maintaining BGP Dynamic Update Groups	102
Troubleshooting Tips	103
Configuration Examples for a Basic BGP Network	104
Example Configuring a BGP Process and Customizing Peers	104

Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers	105
Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number	107
Example NLRI to AFI Configuration	108
Examples Removing BGP Configuration Commands Using a Redistribution Example	110
Examples BGP Soft Reset	111
Example Resetting BGP Peers Using 4-Byte Autonomous System Numbers	111
Example Resetting and Displaying Basic BGP Information	112
Examples Aggregating Prefixes Using BGP	113
Example Configuring a BGP Peer Group	114
Example Configuring Peer Session Templates	115
Example Configuring Peer Policy Templates	115
Examples Monitoring and Maintaining BGP Dynamic Update Peer-Groups	116
Where to Go Next	117
Additional References	117
Feature Information for Configuring a Basic BGP Network	119
Connecting to a Service Provider Using External BGP	125
Finding Feature Information	125
Prerequisites for Connecting to a Service Provider Using External BGP	125
Restrictions for Connecting to a Service Provider Using External BGP	126
Information About Connecting to a Service Provider Using External BGP	126
External BGP Peering	126
BGP Autonomous System Number Formats	128
BGP Attributes	130
Multihoming	131
MED Attribute	132
Transit Versus Nontransit Traffic	132
BGP Policy Configuration	132
BGP Prefix-Based Outbound Route Filtering	133
BGP Communities	134
Extended Communities	134
Extended Community Lists	135
Administrative Distance	136
BGP Route Map Policy Lists	136

BGP Route Map with a Continue Clause	136
Route Map Operation Without Continue Clauses	137
Route Map Operation with Continue Clauses	137
Match Operations with Continue Clauses	137
Set Operations with Continue Clauses	137
Restrictions	138
How to Connect to a Service Provider Using External BGP	138
Influencing Inbound Path Selection	138
Influencing Inbound Path Selection by Modifying the AS-path Attribute	138
Influencing Inbound Path Selection by Setting the MED Attribute	143
Influencing Outbound Path Selection	146
Influencing Outbound Path Selection Using the Local_Pref Attribute	147
Filtering Outbound BGP Route Prefixes	150
Configuring BGP Peering with ISPs	153
Configuring Multihoming with Two ISPs	153
Multihoming with a Single ISP	157
Configuring Multihoming to Receive the Full Internet Routing Table	165
Configuring BGP Policies	169
Filtering BGP Prefixes with Prefix Lists	169
Filtering BGP Prefixes with AS-path Filters	173
Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers	176
Filtering Traffic Using Community Lists	180
Filtering Traffic Using Extended Community Lists	185
Filtering Traffic Using a BGP Route Map Policy List	189
Filtering Traffic Using Continue Clauses in a BGP Route Map	193
Configuration Examples for Connecting to a Service Provider Using External BGP	196
Influencing Inbound Path Selection Examples	197
Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example	197
Influencing Outbound Path Selection Examples	199
Filtering BGP Prefixes with Prefix Lists Examples	200
Filtering BGP Prefixes Using a Single Prefix List	200
Filtering BGP Prefixes Using a Group of Prefixes	201
Adding or Deleting Prefix List Entries	201
Filtering Traffic Using Community Lists Examples	201

Filtering Traffic Using AS-path Filters Example	202
Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers Examples	202
Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example	203
Filtering Traffic Using a BGP Route Map Example	206
Filtering Traffic Using Continue Clauses in a BGP Route Map Examples	206
Where to Go Next	207
Additional References	208
Feature Information for Connecting to a Service Provider Using External BGP	209
Configuring BGP Neighbor Session Options	215
Finding Feature Information	215
Prerequisites for Configuring BGP Neighbor Session Options	215
Restrictions for Configuring BGP Neighbor Session Options	216
Information About Configuring BGP Neighbor Session Options	216
BGP Neighbor Sessions	216
BGP Support for Fast Peering Session Deactivation	216
BGP Hold Timer	216
BGP Fast Peering Session Deactivation	216
Selective Address Tracking for BGP Fast Session Deactivation	217
BFD Support of BGP IPv6 Neighbors	217
BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached	217
Prefix Limits and BGP Peering Sessions	217
BGP Neighbor Session Restart with the Maximum Prefix Limit	218
BGP Network Autonomous System Migration	218
Autonomous System Migration for BGP Networks	218
Dual Autonomous System Support for BGP Network Autonomous System Migration	218
BGP Network Migration to 4-Byte Autonomous System Numbers	219
TTL Security Check for BGP Neighbor Sessions	219
BGP Support for the TTL Security Check	219
TTL Security Check for BGP Neighbor Sessions	220
TTL Security Check Support for Multihop BGP Neighbor Sessions	220
Benefits of the BGP Support for TTL Security Check	220
BGP Support for TCP Path MTU Discovery per Session	221
Path MTU Discovery	221

BGP Neighbor Session TCP PMTUD	221
BGP Dynamic Neighbors	222
How to Configure BGP Neighbor Session Options	222
Configuring Fast Session Deactivation	222
Configuring Fast Session Deactivation for a BGP Neighbor	222
Configuring Selective Address Tracking for Fast Session Deactivation	224
What to Do Next	226
Configuring BFD for BGP IPv6 Neighbors	226
Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded	229
Troubleshooting Tips	232
Configuring Dual-AS Peering for Network Migration	233
Configuring the TTL Security Check for BGP Neighbor Sessions	235
Configuring BGP Support for TCP Path MTU Discovery per Session	239
Disabling TCP Path MTU Discovery Globally for All BGP Sessions	239
Disabling TCP Path MTU Discovery for a Single BGP Neighbor	241
Enabling TCP Path MTU Discovery Globally for All BGP Sessions	244
Enabling TCP Path MTU Discovery for a Single BGP Neighbor	246
Implementing BGP Dynamic Neighbors Using Subnet Ranges	248
Configuration Examples for BGP Neighbor Session Options	255
Example Configuring Fast Session Deactivation for a BGP Neighbor	255
Example Configuring Selective Address Tracking for Fast Session Deactivation	256
Example Configuring BFD for a BGP IPv6 Neighbor	256
Example Restart Session After Maximum Number of Prefixes From Neighbor Reached	256
Configuration Examples for Dual AS Peering for Network Migration	256
Example: Dual AS Configuration	256
Example: Dual AS Confederation Configuration	257
Example: Replace an AS with Another AS in Routing Updates	257
Example Configuring the TTL-Security Check	258
Examples Configuring BGP Support for TCP Path MTU Discovery per Session	258
Example Disabling TCP Path MTU Discovery Globally for All BGP Sessions	258
Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor	258
Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions	259
Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor	259
Example Implementing BGP Dynamic Neighbors Using Subnet Ranges	259

Where to Go Next	261
Additional References	261
Feature Information for Configuring BGP Neighbor Session Options	263
Configuring Internal BGP Features	269
Finding Feature Information	269
Information About Internal BGP Features	269
BGP Routing Domain Confederation	269
BGP Route Reflector	270
Route Reflector Mechanisms to Avoid Routing Loops	273
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer	273
BGP VPLS Autodiscovery Support on Route Reflector	274
BGP Route Dampening	274
Route Dampening Minimizes Route Flapping	274
BGP Route Dampening Terms	274
How to Configure Internal BGP Features	275
Configuring a Routing Domain Confederation	275
Configuring a Route Reflector	276
Configuring a Route Reflector Using a Route Map to Set Next Hop for iBGP Peer	276
Adjusting BGP Timers	280
Configuring the Router to Consider a Missing MED as Worst Path	281
Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths	281
Configuring the Router to Use the MED to Choose a Path in a Confederation	281
Enabling BGP Route Dampening	282
Monitoring and Maintaining BGP Route Dampening	282
Internal BGP Feature Configuration Examples	284
Example BGP Confederation Configurations with Route Maps	284
Examples BGP Confederation	284
Example Route Reflector Using a Route Map to Set Next Hop for iBGP Peer	285
Example BGP VPLS Autodiscovery Support on Route Reflector	286
Additional References	286
Feature Information for Configuring Internal BGP Features	288
Configuring Advanced BGP Features	291
Finding Feature Information	291
Prerequisites for Configuring Advanced BGP Features	291

Restrictions for Configuring Advanced BGP Features	291
Information About Configuring Advanced BGP Features	292
BGP Version 4	292
BGP Support for Next-Hop Address Tracking	292
BGP Next-Hop Address Tracking	292
Default BGP Scanner Behavior	293
Selective BGP Next-Hop Route Filtering	293
BGP Next_Hop Attribute	293
BGP Nonstop Forwarding Awareness	293
Cisco NSF Routing and Forwarding Operation	294
Cisco Express Forwarding for NSF	294
BGP Graceful Restart for NSF	295
BGP NSF Awareness	295
BGP Graceful Restart per Neighbor	296
BGP Peer Session Templates	296
BGP Route Dampening	297
BFD for BGP	298
BGP MIB Support	298
BGP Support for MTR	300
BGP Network Scope	300
MTR CLI Hierarchy Under BGP	301
BGP Sessions for Class-Specific Topologies	301
Topology Translation Using BGP	302
Topology Import Using BGP	302
How to Configure Advanced BGP Features	302
Configuring BGP Next-Hop Address Tracking	302
Disabling BGP Next-Hop Address Tracking	302
Adjusting the Delay Interval for BGP Next-Hop Address Tracking	304
Configuring BGP Selective Next-Hop Route Filtering	305
Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart	309
Enabling BGP Global NSF Awareness Using BGP Graceful Restart	309
Troubleshooting Tips	310
What to Do Next	311
Configuring BGP NSF Awareness Timers	311
What to Do Next	312

Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates	313
Enabling BGP Graceful Restart for an Individual BGP Neighbor	318
Disabling BGP Graceful Restart for a BGP Peer Group	320
Verifying the Configuration of BGP Nonstop Forwarding Awareness	323
Configuring BGP Route Dampening	324
Enabling and Configuring BGP Route Dampening	324
Monitoring and Maintaining BGP Route Dampening	326
Decreasing BGP Convergence Time Using BFD	327
Prerequisites	328
Restrictions	328
Configuring BFD Session Parameters on the Interface	328
Configuring BFD Support for BGP	329
Monitoring and Troubleshooting BFD for Cisco 7600 Series Routers	331
What to Do Next	331
Enabling BGP MIB Support	331
Configuring BGP Support for MTR	332
Activating an MTR Topology Using BGP	333
What to Do Next	337
Importing Routes from an MTR Topology Using BGP	338
Where to Go Next	341
Additional References	341
Feature Information for Configuring Advanced BGP Features	342
Configuring Multiprotocol BGP (MP-BGP) Support for CLNS	349
Finding Feature Information	349
Restrictions for Configuring MP-BGP Support for CLNS	349
Information About Configuring MP-BGP Support for CLNS	350
Address Family Routing Information	350
Design Features of MP-BGP Support for CLNS	350
Generic BGP CLNS Network Topology	350
DCN Network Topology	352
Benefits of MP-BGP Support for CLNS	353
How to Configure MP-BGP Support for CLNS	353
Configuring and Activating a BGP Neighbor to Support CLNS	354
Configuring an IS-IS Routing Process	356
Configuring Interfaces That Connect to BGP Neighbors	357

Configuring Interfaces Connected to the Local OSI Routing Domain	359
Advertising Networking Prefixes	360
Redistributing Routes from BGP into IS-IS	363
Redistributing Routes from IS-IS into BGP	364
Configuring BGP Peer Groups and Route Reflectors	366
Filtering Inbound Routes Based on NSAP Prefixes	368
Filtering Outbound BGP Updates Based on NSAP Prefixes	369
Originating Default Routes for a Neighboring Routing Domain	372
Verifying MP-BGP Support for CLNS	374
Troubleshooting MP-BGP Support for CLNS	376
Configuration Examples for MP-BGP Support for CLNS	377
Configuring and Activating a BGP Neighbor to Support CLNS Example	378
Configuring an IS-IS Routing Process Example	378
Configuring Interfaces Example	378
Advertising Networking Prefixes Example	378
Redistributing Routes from BGP into IS-IS Example	379
Redistributing Routes from IS-IS into BGP Example	379
Configuring BGP Peer Groups and Route Reflectors Example	379
Filtering Inbound Routes Based on NSAP Prefixes Example	380
Filtering Outbound BGP Updates Based on NSAP Prefixes Example	380
Originating a Default Route and Outbound Route Filtering Example	380
Implementing MP-BGP Support for CLNS Example	381
Autonomous System AS65101	382
Autonomous System AS65202	382
Autonomous System AS65303	383
Autonomous System AS65404	384
Additional References	386
Feature Information for Configuring MP-BGP Support for CLNS	387
Glossary	389
BGP Link Bandwidth	391
Finding Feature Information	391
Prerequisites for BGP Link Bandwidth	391
Restrictions for BGP Link Bandwidth	392
Information About BGP Link Bandwidth	392
BGP Link Bandwidth Overview	392

Link Bandwidth Extended Community Attribute	392
Benefits of the BGP Link Bandwidth Feature	392
How to Configure BGP Link Bandwidth	393
Configuring and Verifying BGP Link Bandwidth	393
Configuration Examples for BGP Link Bandwidth	395
Example: BGP Link Bandwidth Configuration	395
Example: Verifying BGP Link Bandwidth	398
Where to Go Next	399
Additional References	399
Feature Information for BGP Link Bandwidth	400
iBGP Multipath Load Sharing	403
Finding Feature Information	403
Restrictions for iBGP Multipath Load Sharing	403
Information about iBGP Multipath Load Sharing	404
iBGP Multipath Load Sharing Overview	404
Benefits of iBGP Multipath Load Sharing	405
How To Configure iBGP Multipath Load Sharing	405
Configuring iBGP Multipath Load Sharing	405
Verifying iBGP Multipath Load Sharing	406
Monitoring and Maintaining iBGP Multipath Load Sharing	408
Configuration Examples for iBGP Multipath Load Sharing	408
Example iBGP Multipath Load Sharing in a Non-MPLS Topology	409
Example iBGP Multipath Load Sharing in an MPLS VPN Topology	409
Additional References	410
Command Reference	411
Feature Information for iBGP Multipath Load Sharing	411
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	413
Finding Feature Information	413
Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	414
Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	414
Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	414
Multipath Load Sharing Between eBGP and iBGP	414
eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network	415
eBGP and iBGP Multipath Load Sharing With Route Reflectors	416
Benefits of Multipath Load Sharing for Both eBGP and iBGP	416

How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	416
Configuring Multipath Load Sharing for Both eBGP and iBGP	416
Verifying Multipath Load Sharing for Both eBGP and iBGP	418
Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature	419
eBGP and iBGP Multipath Load Sharing Configuration Example	419
eBGP and iBGP Multipath Load Sharing Verification Examples	419
Where to Go Next	420
Additional References	420
Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	422
Loadsharing IP Packets Over More Than Six Parallel Paths	423
Finding Feature Information	423
Overview of Loadsharing IP Packets over More Than Six Parallel Paths	423
Additional References	424
Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths	425
BGP Policy Accounting	427
Finding Feature Information	427
Prerequisites	427
Information About BGP Policy Accounting	427
BGP Policy Accounting Overview	427
Benefits of BGP Policy Accounting	428
How to Configure BGP Policy Accounting	429
Specifying the Match Criteria for BGP Policy Accounting	429
Classifying the IP Traffic and Enabling BGP Policy Accounting	430
Verifying BGP Policy Accounting	431
Monitoring and Maintaining BGP Policy Accounting	432
Configuration Examples	432
Specifying the Match Criteria for BGP Policy Accounting Example	432
Classifying the IP Traffic and Enabling BGP Policy Accounting Example	433
Additional References	433
Feature Information for BGP Policy Accounting	434
BGP Cost Community	437
Finding Feature Information	437
Prerequisites for the BGP Cost Community Feature	437
Restrictions for the BGP Cost Community Feature	437
Information About the BGP Cost Community Feature	438

BGP Cost Community Overview	438
How the BGP Cost Community Influences the Best Path Selection Process	438
Cost Community Support for Aggregate Routes and Multipaths	439
Influencing Route Preference in a Multi-Exit IGP Network	440
BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links	440
How to Configure the BGP Cost Community Feature	441
Configuring the BGP Cost Community	441
Verifying the Configuration of the BGP Cost Community	443
Troubleshooting Tips	443
Configuration Examples for the BGP Cost Community Feature	444
BGP Cost Community Configuration Example	444
BGP Cost Community Verification Examples	444
Where to Go Next	445
Additional References	445
Command Reference	447
Feature Information for BGP Cost Community	447
BGP Support for IP Prefix Import from Global Table into a VRF Table	449
Finding Feature Information	449
Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table	449
Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table	450
Information About BGP Support for IP Prefix Import from Global Table into a VRF Table	450
Importing IPv4 Prefixes into a VRF	450
Black Hole Routing	450
Classifying Global Traffic	450
Unicast Reverse Path Forwarding	451
How to Import IP Prefixes from Global Table into a VRF Table	451
Defining IPv4 IP Prefixes to Import	451
Creating the VRF and the Import Route Map	452
Filtering on the Ingress Interface	455
Verifying Global IP Prefix Import	456
Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table	457
Configuring Global IP Prefix Import Example	457
Verifying Global IP Prefix Import Example	458
Additional References	459

Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table	460
BGP per Neighbor SoO Configuration	463
Finding Feature Information	463
Prerequisites for BGP per Neighbor SoO Configuration	463
Restrictions for BGP per Neighbor SoO Configuration	463
Information About Configuring BGP per Neighbor SoO	464
Site of Origin BGP Community Attribute	464
Route Distinguisher	464
BGP Support for 4-Byte Autonomous System Numbers	464
BGP per Neighbor Site of Origin Configuration	465
Benefits of BGP per Neighbor Site of Origin	465
How to Configure BGP per Neighbor SoO	466
Enabling Cisco Express Forwarding and Configuring VRF Instances	466
Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template	469
BGP Peer Policy Templates	469
Configuring a per Neighbor SoO Value Using a BGP neighbor Command	472
Configuring a per Neighbor SoO Value Using a BGP Peer Group	474
Configuration Examples for BGP per Neighbor SoO Configuration	477
Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template Example	477
Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template Example	478
Configuring a per Neighbor SoO Value Using a BGP neighbor Command Example	478
Configuring a per Neighbor SoO Value Using a BGP neighbor Command and 4-Byte Autonomous System Numbers Example	479
Configuring a per Neighbor SoO Value Using a BGP Peer Group Example	479
Where to Go Next	479
Additional References	480
Feature Information for BGP per Neighbor SoO Configuration	481
Per-VRF Assignment of BGP Router ID	485
Finding Feature Information	485
Prerequisites for Per-VRF Assignment of BGP Router ID	485
Information About Per-VRF Assignment of BGP Router ID	485
BGP Router ID	486
Per-VRF Router ID Assignment	486
Route Distinguisher	486
How to Configure Per-VRF Assignment of BGP Router ID	486

Configuring VRF Instances	486
Associating VRF Instances with Interfaces	488
Manually Configuring a BGP Router ID per VRF	491
Automatically Assigning a BGP Router ID per VRF	496
Configuration Examples for Per-VRF Assignment of BGP Router ID	503
Manually Configuring a BGP Router ID per VRF Examples	503
Automatically Assigning a BGP Router ID per VRF Examples	505
Additional References	509
Command Reference	510
Feature Information for Per-VRF Assignment of BGP Router ID	511
BGP Next Hop Unchanged	513
Finding Feature Information	513
Information About BGP Next Hop Unchanged	513
BGP Next Hop Unchanged	513
How to Configure BGP Next Hop Unchanged	514
Configuring the BGP Next Hop Unchanged for an eBGP Peer	515
Configuration Example for BGP Next Hop Unchanged	516
Example BGP Next Hop Unchanged for an eBGP Peer	517
Additional References	517
Feature Information for BGP Next Hop Unchanged	517
BGP Support for the L2VPN Address Family	519
Finding Feature Information	519
Prerequisites for BGP Support for the L2VPN Address Family	519
Restrictions for BGP Support for the L2VPN Address Family	519
Information About BGP Support for the L2VPN Address Family	520
L2VPN Address Family	520
VPLS ID	521
How to Configure BGP Support for the L2VPN Address Family	521
Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family	521
Examples	527
What to Do Next	527
Configuration Examples for BGP Support for the L2VPN Address Family	527
Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example	527
Where to Go Next	530
Additional References	530

Feature Information for BGP Support for the L2VPN Address Family	531
BGP 4 MIB Support for per-Peer Received Routes	533
Finding Feature Information	533
Restrictions on BGP 4 MIB Support for Per-Peer Received Routes	533
Information About BGP 4 MIB Support for Per-Peer Received Routes	534
BGP 4 MIB Support for Per-Peer Received Routes Overview	534
BGP 4 per-Peer Received Routes Table Elements and Objects	535
MIB Tables and Objects	535
AFIs and SAFIs	536
Network Address Prefix Descriptions for the NLRI Field	536
Benefits of BGP 4 MIB Support for Per-Peer Received Routes	537
Additional References	538
Feature Information for BGP 4 MIB Support for per-Peer Received Routes	539
Glossary	539
BGP Event-Based VPN Import	541
Finding Feature Information	541
Prerequisites for BGP Event-Based VPN Import	541
Information About BGP Event-Based VPN Import	541
BGP Event-Based VPN Import	542
Import Path Selection Policy	542
Import Path Limit	542
How to Configure BGP Event-Based VPN Import	543
Configuring a Multiprotocol VRF	543
Configuring Event-Based VPN Import Processing for BGP Paths	546
Monitoring and Troubleshooting BGP Event-Based VPN Import Processing	547
Configuration Examples for BGP Event-Based VPN Import	549
Configuring Event-Based VPN Import Processing for BGP Paths Example	549
Where to Go Next	550
Additional References	550
Feature Information for BGP Event-Based VPN Import	551
BGP Best External	553
Finding Feature Information	553
Contents	553
Prerequisites for BGP Best External	554
Restrictions for BGP Best External	554

Information About BGP Best External	554
BGP Best External Overview	554
What the Best External Route Means	555
How the BGP Best External Feature Works	555
Configuration Modes for Enabling BGP Best External	556
How to Configure BGP Best External	556
Enabling the BGP Best External Feature	556
Verifying the BGP Best External Feature	559
Configuration Examples for BGP Best External	561
Example Configuring the BGP Best External Feature	561
Additional References	562
Feature Information for BGP Best External	563
BGP PIC Edge for IP and MPLS-VPN	565
Finding Feature Information	565
Contents	565
Prerequisites for BGP PIC	566
Restrictions for BGP PIC	566
Information About BGP PIC	566
Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature	567
How BGP Converges Under Normal Circumstances	567
How BGP PIC Improves Convergence	567
BGP Fast Reroute's Role in the BGP PIC Feature	568
How a Failure Is Detected	569
How BGP PIC Achieves Subsecond Convergence	569
How BGP PIC Improves Upon the Functionality of MPLS VPN--BGP Local Convergence	570
Configuration Modes for Enabling BGP PIC	570
BGP PIC Scenarios	570
IP PE-CE Link and Node Protection on the CE Side (Dual PEs)	570
IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes)	571
IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path	572
IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path	573
Cisco Express Forwarding Recursion	574
How to Configure BGP PIC	574
Configuring BGP PIC	575

Configuration Examples for BGP PIC **577**

Example: Configuring BGP PIC **578**

Example: Displaying Backup Alternate Paths for BGP PIC **579**

Additional References **580**

Feature Information for BGP PIC **582**



Cisco BGP Overview

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

- [Finding Feature Information, page 1](#)
- [Prerequisites for Cisco BGP, page 1](#)
- [Restrictions for Cisco BGP, page 1](#)
- [Information About Cisco BGP, page 2](#)
- [Where to Go Next, page 15](#)
- [Additional References, page 15](#)
- [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 feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Cisco BGP

This document assumes knowledge of 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 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 concurrent BGP address family and subaddress family configurations.

Information About Cisco BGP

- [BGP Version 4 Functional Overview, page 2](#)
- [BGP Autonomous Systems, page 3](#)
- [BGP Autonomous System Number Formats, page 4](#)
- [Classless Interdomain Routing, page 6](#)
- [Multiprotocol BGP, page 7](#)
- [Benefits of Using Multiprotocol BGP Versus BGP, page 7](#)
- [Multiprotocol BGP Extensions for IP Multicast, page 7](#)
- [NLRI Configuration CLI, page 9](#)
- [Cisco BGP Address Family Model, page 10](#)
- [IPv4 Address Family, page 12](#)
- [IPv6 Address Family, page 12](#)
- [CLNS Address Family, page 12](#)
- [VPNv4 Address Family, page 13](#)
- [L2VPN Address Family, page 13](#)
- [BGP CLI Removal Considerations, page 14](#)

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 assigned a random port number. Cisco IOS software supports BGP version 4 and it is this version that has been used by Internet Service Providers to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use. RFC 2858 introduced multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IPv4, 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. For more details about configuring BGP peer sessions and other tasks to build a basic BGP network, see the "Configuring a Basic BGP Network" module.

BGP uses a path-vector routing algorithm to exchange network reachability information with other BGP speaking networking devices. Network reachability information is exchanged between BGP peers in routing updates. Network reachability information contains the network number, path specific attributes, and the list of autonomous system numbers that a route must transit through to reach a destination network. This list is contained in the AS-path attribute. BGP prevents routing loops by rejecting any routing update that contains the local autonomous system number because this indicates that the route has already travelled through that autonomous system and a loop would therefore be created. The BGP path-vector routing algorithm is a combination of the distance-vector routing algorithm and the AS-path loop detection. For

more details about configuration tasks to configure various options involving BGP neighbor peer sessions, see the "Configuring BGP Neighbor Session Options" module.

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

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

BGP Autonomous Systems

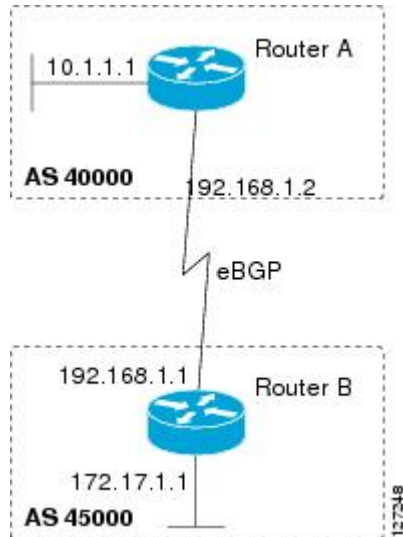
An autonomous system is a network controlled by a single technical administration entity. BGP autonomous systems are used to divide global external networks into individual routing domains where local routing policies are applied. This organization simplifies routing domain administration and simplifies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

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

The figure below illustrates two routers in separate autonomous systems that can be connected using BGP. Router A and Router B are Internet service provider (ISP) routers in separate routing domains that use

public autonomous system numbers. These routers carry traffic across the Internet. Router A and Router B are connected through eBGP peering sessions.

Figure 1 BGP Topology with Two Autonomous Systems



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

BGP Autonomous System Number Formats

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

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

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

Asdot Only Autonomous System Number Formatting

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)SXI1, 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 IOS software. Enabling CIDR affects how packets are forwarded but it does not change the operation of BGP.

Multiprotocol BGP

Cisco IOS 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 dedicated to multicast traffic, perhaps to limit which resources are used for which traffic. For example, you want all multicast traffic exchanged at one network access point (NAP). Multiprotocol BGP allows you to have a unicast routing topology different from a multicast routing topology that allows you more control over your network and resources.

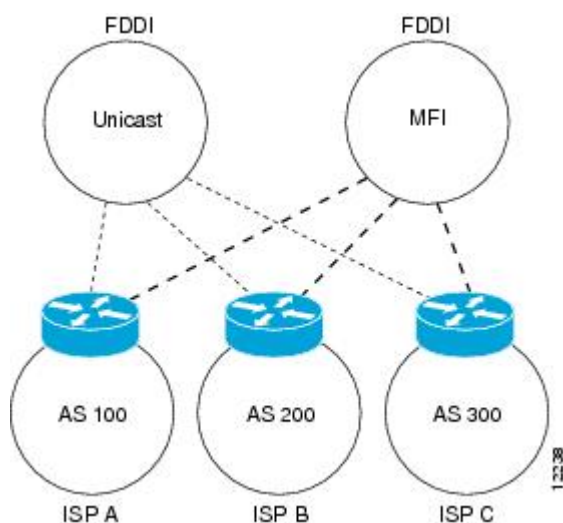
In BGP, the only way to perform interdomain multicast routing is to use the BGP infrastructure that is in place for unicast routing. If the routers are not multicast-capable, or there are differing policies about where multicast traffic should flow, multicast routing cannot be supported without multiprotocol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

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

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

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

Figure 2 *Incongruent Unicast and Multicast Routes*

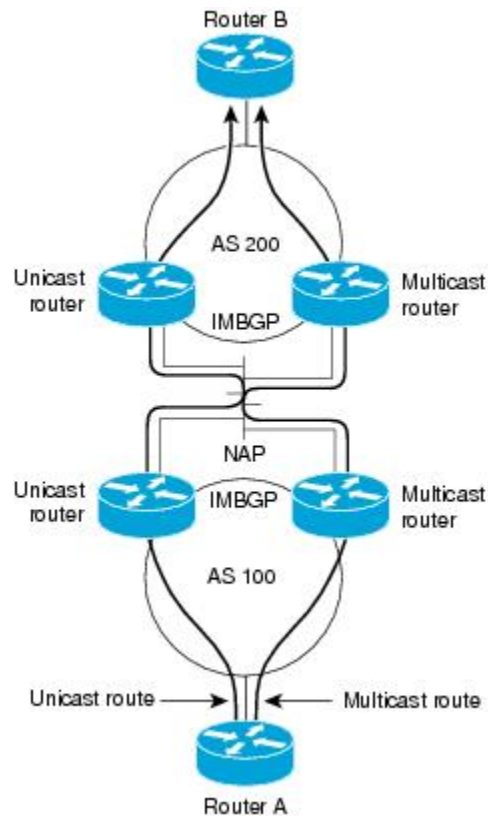


The figure below is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they do not support or are not configured to perform multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both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

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

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

Figure 3 **Multicast BGP Environment**



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

NLRI Configuration CLI

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

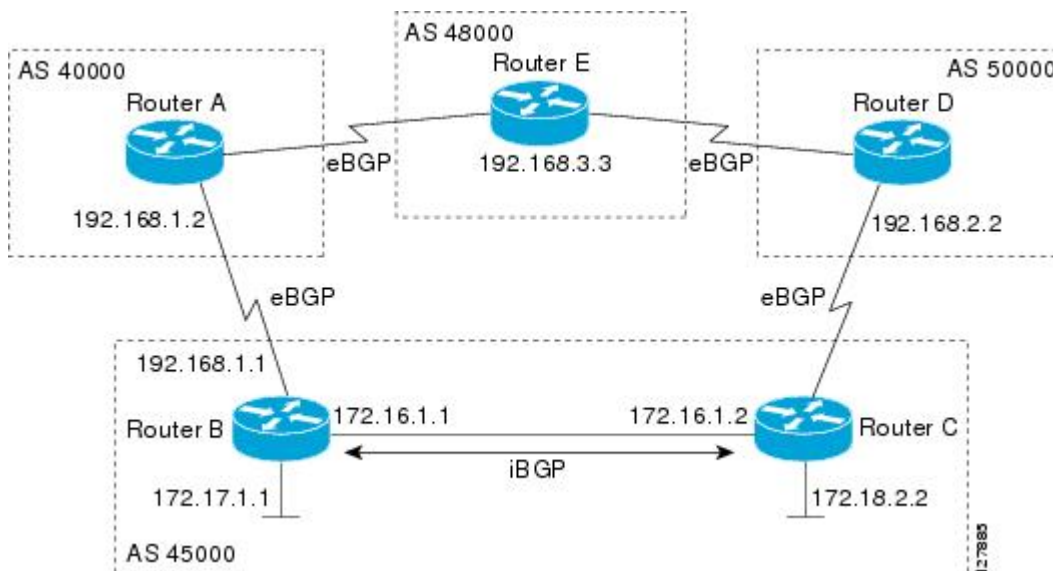
Using the BGP hybrid CLI feature, you can configure commands in the address family VPNv4 format and save these command configurations without modifying an existing NLRI formatted configuration. If you want to use other address family configurations such as IPv4 unicast or multicast, then you must upgrade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

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

Cisco BGP Address Family Model

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

Figure 4 BGP Network Topology for Multiple Address Families



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
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

SAFI Field Value	Description	Reference
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

In Cisco IOS Release 12.2(33)SRB and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or Generic Routing Encapsulation (GRE). The L2VPN address family is configured under BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

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

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS Virtual Private LAN Service (VPLS) feature. VPLS enables flexibility in deploying

services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the "VPLS Autodiscovery: BGP Based" feature.

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" feature in Cisco IOS Release 12.2(33)SRB.

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 results in your network. When you remove an access list or route map, you must also review the commands that referenced that access list or route map to consider whether the command will give you the behavior you intended.

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

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

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

Where to Go Next

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Configuring basic BGP tasks	"Configuring a Basic BGP Network" module

Related Topic	Document Title
Configuring BGP neighbor session options	"Configuring BGP Neighbor Session Options" module
Configuring BGP to connect to a service provider	"Connecting to a Service Provider Using External BGP" module
Configuring internal BGP (iBGP) tasks	"Configuring Internal BGP Features" module
Configuring advanced BGP features	"Configuring Advanced BGP Features" module
Configuring Multiprotocol BGP with CLNS	"Configuring Multiprotocol BGP (MP-BGP) Support for CLNS" module
Configuring basic IP multicast tasks	"Configuring Basic IP Multicast" module

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	A Border Gateway Protocol 4 (BGP-4)
RFC 4893	BGP Support for Four-Octet AS Number Space
RFC 5396	Textual Representation of Autonomous System (AS) Numbers
RFC 5398	Autonomous System (AS) Number Reservation for Documentation Use

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/cisco/web/support/index.html</p>

Feature Information for Cisco BGP Overview

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

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

Table 5 *Feature Information for Cisco BGP Overview*

Feature Name	Releases	Feature Information
BGP Support for 4-Byte ASN	12.0(32)S12 12.0(32)SY8 12.0(33)S3 12.2(33)SRE 12.2(33)XNE 12.2(33)SXI1 12.4(24)T 15.0(1)S Cisco IOS XE 3.1.0SG	<p>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.</p> <p>In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, and 12.2(33)SXI1, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the bgp asnotation dot command.</p> <p>In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses asdot as the only configuration format, regular expression match, and output display, with no asplain support.</p> <p>The following commands were introduced or modified by this feature: bgp asnotation dot, bgp confederation identifier, bgp confederation peers, all clear ip bgp commands that configure an autonomous system number, ip as-path access-list, ip extcommunity-list, match</p>

Feature Name	Releases	Feature Information
BGP Support for the L2VPN Address Family	12.2(33)SRB	<p data-bbox="1151 289 1516 569">source-protocol, neighbor local-as, neighbor remote-as, neighbor so0, redistribute (IP), router bgp, route-target, set as-path, set extcommunity, set origin, so0, all show ip bgp commands that display an autonomous system number, and show ip extcommunity-list.</p> <p data-bbox="1151 600 1516 1167">BGP Support for the L2VPN address family introduced a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN routing information base (RIB) to store endpoint provisioning information which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a Pseudowire mesh to support L2VPN-based services.</p> <p data-bbox="1151 1188 1516 1310">The following commands were introduced or modified by this feature: address-family l2vpn, show ip bgp l2vpn.</p>

Feature Name	Releases	Feature Information
Configuring Multiprotocol BGP Support for CLNS	12.2(33)SRB	<p>The Multiprotocol BGP (MP-BGP) Support for CLNS feature provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks.</p> <p>The following commands were introduced or modified by this feature: clear bgp nsap, clear bgp nsap dampening, clear bgp nsap external, clear bgp nsap flap-statistics, clear bgp nsap peer-group, debug bgp nsap, debug bgp nsap dampening, debug bgp nsap updates, neighbor prefix-list, network (BGP and multiprotocol BGP), redistribute (BGP to ISO ISIS), redistribute (ISO ISIS to BGP), show bgp nsap, show bgp nsap community, show bgp nsap community-list, show bgp nsap dampened-paths, show bgp nsap filter-list, show bgp nsap flap-statistics, show bgp nsap inconsistent-as, show bgp nsap neighbors, show bgp nsap paths, show bgp nsap quote-regexp, show bgp nsap regexp, show bgp nsap summary.</p>

Feature Name	Releases	Feature Information
Multiprotocol BGP	Cisco IOS XE 3.1.0SG	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.

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: 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)

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



Configuring a Basic BGP Network

This module describes the basic tasks to configure a basic Border Gateway Protocol (BGP) network. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. The Cisco IOS 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 23](#)
- [Prerequisites for Configuring a Basic BGP Network, page 23](#)
- [Restrictions for Configuring a Basic BGP Network, page 23](#)
- [Information About Configuring a Basic BGP Network, page 24](#)
- [How to Configure a Basic BGP Network, page 39](#)
- [Configuration Examples for a Basic BGP Network, page 104](#)
- [Where to Go Next, page 117](#)
- [Additional References, page 117](#)
- [Feature Information for Configuring a Basic BGP Network, page 119](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Configuring a Basic BGP Network

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

Restrictions for Configuring a Basic BGP Network

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

Information About Configuring a Basic BGP Network

- [BGP Version 4, page 24](#)
- [BGP Router ID, page 25](#)
- [BGP-Speaker and Peer Relationships, page 25](#)
- [BGP Autonomous System Number Formats, page 25](#)
- [Cisco Implementation of 4-Byte Autonomous System Numbers, page 28](#)
- [BGP Peer Session Establishment, page 29](#)
- [Cisco Implementation of BGP Global and Address Family Configuration Commands, page 29](#)
- [BGP Session Reset, page 31](#)
- [BGP Route Aggregation, page 31](#)
- [BGP Aggregation Route AS-SET Information Generation, page 32](#)
- [Routing Policy Change Management, page 32](#)
- [Conditional BGP Route Injection, page 33](#)
- [BGP Peer Groups, page 34](#)
- [BGP Backdoor Routes, page 34](#)
- [Peer Groups and BGP Update Messages, page 35](#)
- [BGP Update Group, page 35](#)
- [BGP Dynamic Update Group Configuration, page 35](#)
- [BGP Peer Templates, page 35](#)
- [Inheritance in Peer Templates, page 36](#)
- [Peer Session Templates, page 37](#)
- [Peer Policy Templates, page 38](#)
- [BGP IPv6 Neighbor Activation Under the IPv4 Address Family, page 39](#)

BGP Version 4

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

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

**Note**

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

BGP Router ID

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

BGP-Speaker and Peer Relationships

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

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

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

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

BGP Autonomous System Number Formats

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

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

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

Asdot Only Autonomous System Number Formatting

In Cisco IOS 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 6 *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 7 **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 8 **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)SXII, 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.

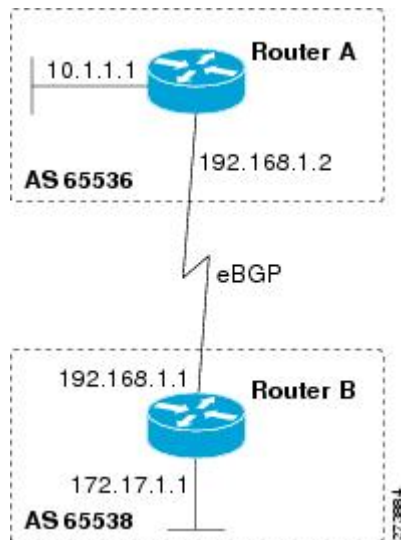
Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous 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 5 BGP Peers in Two Autonomous Systems Using 4-Byte Numbers



BGP Peer Session Establishment

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

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

Cisco Implementation of BGP Global and Address Family Configuration Commands

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

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

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

Table 9 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
```

In Cisco IOS Releases 12.0(22)S, 12.2(15)T, and later releases, the **bgp upgrade-cli** command simplifies the migration of BGP networks and existing configurations from the network layer reachability information (NLRI) format to the address family format. Network operators can configure commands in the address family identifier (AFI) format and save these command configurations to existing NLRI formatted configurations. The BGP hybrid command-line interface (CLI) does not add support for complete AFI and NLRI integration because of the limitations of the NLRI format. For complete support of AFI commands and features, we recommend upgrading existing NLRI configurations with the **bgp upgrade-cli** command.

For a configuration example of migrating BGP configurations from the NLRI format to the address family format, see .

BGP Session Reset

Whenever there is a change in the routing policy due to a configuration change, BGP peering sessions must be reset using the **clear ip bgp** command. Cisco IOS software support the following three mechanisms to reset BGP peering sessions:

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

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

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

BGP Route Aggregation

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

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

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

BGP 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 10 *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.

Type of Reset	Advantages	Disadvantages
Configured inbound soft reset (uses the neighbor soft-reconfiguration router configuration command)	<p>Can be used when both BGP routers do not support the automatic route refresh capability.</p> <p>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.</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 routers do not support the automatic route refresh capability.</p> <p>Note Does not reset outbound routing table updates.</p>

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 IOS software provides several methods in which you can originate a prefix into BGP. The existing methods include redistribution and using the **network** or **aggregate-address** command. These methods assume the existence of more specific routing information (matching the route to be originated) in either the routing table or the BGP table.

BGP conditional route injection allows you to originate a prefix into a BGP routing table without the corresponding match. This feature allows more specific routes to be generated based on administrative policy or traffic engineering information in order to provide more specific control over the forwarding of packets to these more specific routes, which are injected into the BGP routing table only if the configured conditions are met. Enabling this feature will allow you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only

prefixes that are equal to or more specific than the original prefix may be injected. BGP conditional route injection is enabled with the **bgp inject-map exist-map** command and uses two route maps (inject map and exist map) to install one (or more) more specific prefixes into a BGP routing table. The exist map specifies the prefixes that the BGP speaker will track. The inject map defines the prefixes that will be created and installed into the local BGP table.

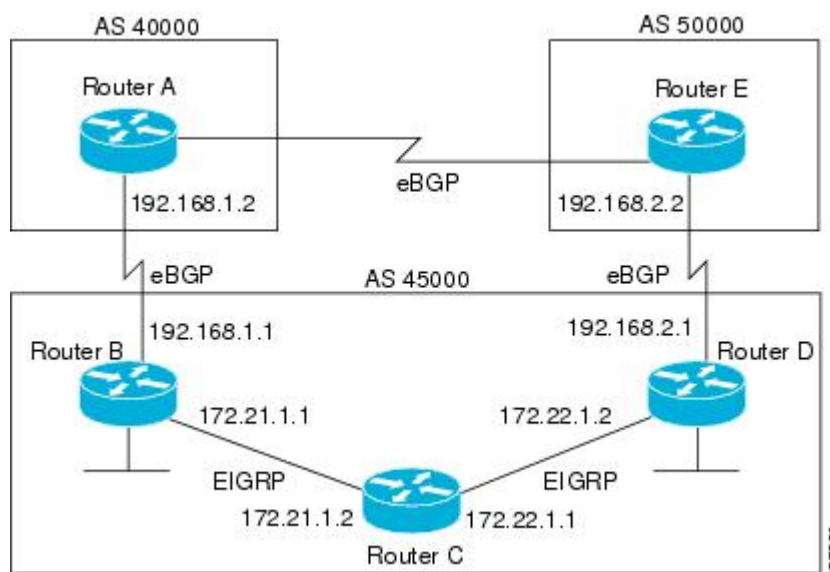
BGP Peer Groups

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

BGP Backdoor Routes

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

Figure 6 BGP Backdoor Route Topology



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 in Cisco IOS Releases 12.0(24)S, 12.2(18)S, 12.3(4)T, or 12.2(27)SBC, 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. In Cisco IOS 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.

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 peer session templates, a peer policy template supports inheritance. However, there are minor differences. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group. Inherited peer policy templates are configured with sequence numbers like route maps. An inherited peer policy template, like a route map, is evaluated starting with the inherit statement with the lowest sequence number and ending with the highest sequence number. However, there is a difference; a peer policy template will not collapse like a route map. Every sequence is evaluated, and if a BGP policy

command is reapplied with a different value, it will overwrite any previous value from a lower sequence number.

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

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

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

BGP IPv6 Neighbor Activation Under the IPv4 Address Family

Prior to Cisco IOS 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, page 40](#)
- [Configuring a BGP Peer, page 43](#)
- [Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers, page 46](#)
- [Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers, page 50](#)
- [Configuring a BGP Peer for the IPv4 VRF Address Family, page 53](#)
- [Customizing a BGP Peer, page 57](#)
- [Removing BGP Configuration Commands Using a Redistribution, page 62](#)
- [Monitoring and Maintaining Basic BGP, page 64](#)
- [Aggregating Route Prefixes Using BGP, page 69](#)
- [Originating BGP Routes, page 79](#)
- [Configuring a BGP Peer Group, page 87](#)
- [Configuring Peer Session Templates, page 89](#)

- [Configuring Peer Policy Templates, page 95](#)
- [Monitoring and Maintaining BGP Dynamic Update Groups, page 102](#)

Configuring a BGP Routing Process

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

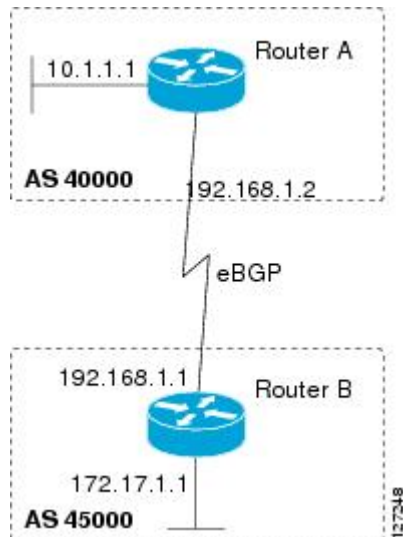


Note

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

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

Figure 7 BGP Topology with Two Autonomous Systems



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *network-number* [**mask** *network-mask*][**route-map** *route-map-name*]
5. **bgp router-id** *ip-address*
6. **timers bgp** *keepalive holdtime*
7. **bgp fast-external-falover**
8. **bgp log-neighbor-changes**
9. **end**
10. **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 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 router to other BGP speakers.
Step 4	network <i>network-number</i> [mask <i>network-mask</i>][route-map <i>route-map-name</i>] Example: Router(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.

Command or Action	Purpose
<p>Step 5 <code>bgp router-id ip-address</code></p> <p>Example:</p> <pre>Router(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 router 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>
<p>Step 6 <code>timers bgp keepalive holdtime</code></p> <p>Example:</p> <pre>Router(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 not receiving a keepalive message that the software declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds.
<p>Step 7 <code>bgp fast-external-fallover</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 8 <code>bgp log-neighbor-changes</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 9 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>
<p>Step 10 <code>show ip bgp [network] [network-mask]</code></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>

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, page 43](#)

Troubleshooting Tips

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

Configuring a BGP Peer

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

Before you perform this task, perform the [Configuring a BGP Routing Process, page 40](#) 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
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></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>
<p>Step 5 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>]</code></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.
<p>Step 6 <code>neighbor <i>ip-address</i> activate</code></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>

Command or Action	Purpose
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
<p>Step 8 <code>show ip bgp [network] [network-mask]</code></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>
<p>Step 9 <code>show ip bgp neighbors [neighbor-address]</code></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 holdtims
Neighbor capabilities:
  Route refresh: advertised and received (old & new)
  Address family IPv4 Unicast: advertised and received
Message statistics:
  InQ depth is 0
  OutQ depth is 0

           Sent           Rcvd
Opens:           1           1
Notifications:  0           0
Updates:         1           2
Keepalives:      13          13
Route Refresh:   0           0
Total:           15          16
Default minimum time between advertisement runs is 30 seconds
```

```

For address family: IPv4 Unicast
BGP table version 13, neighbor version 13/0
Output queue size : 0
Index 1, Offset 0, Mask 0x2
1 update-group member

Prefix activity:
-----
Prefixes Current:      1          1 (Consumes 52 bytes)
Prefixes Total:       1          1
Implicit Withdraw:     0          0
Explicit Withdraw:    0          0
Used as bestpath:     n/a        1
Used as multipath:    n/a        0
-----
Local Policy Denied Prefixes:
-----
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, page 46](#)
- [What to Do Next, page 46](#)

Troubleshooting Tips

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

What to Do Next

If you have BGP peers in a VPN, proceed to the [Configuring a BGP Peer for the IPv4 VRF Address Family, page 53](#). If you do not have BGP peers in a VPN, proceed to the [Customizing a BGP Peer, page 57](#).

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

Perform this task to configure a BGP routing process and BGP peers when the BGP peers are located in an AS that uses 4-byte autonomous system numbers. The address family configured here is the default IPv4

unicast address family, and the configuration is done at Router B in the figure above (in the "Cisco Implementation of 4-Byte Autonomous System Numbers" section). The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

This task requires Cisco IOS Release 12.0(32)SY8, 12.2(33)SXII, or a later release to be running on the router.



Note

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

SUMMARY STEPS

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

DETAILED STEPS

	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>

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(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 autonomous system number, 65538, is defined in asplain notation.
<p>Step 4 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 65536</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"> In this example, the 4-byte autonomous system number, 65536, is defined in asplain notation.
<p>Step 5 Repeat Step 4 to define other BGP neighbors, as required.</p>	<p>--</p>
<p>Step 6 <code>address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</code></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.
<p>Step 7 <code>neighbor <i>ip-address</i> activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.</p>
<p>Step 8 Repeat Step 7 to activate other BGP neighbors, as required.</p>	<p>--</p>
<p>Step 9 <code>network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>]</code></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 10 <code>end</code> Example: Router(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11 <code>show ip bgp [network] [network-mask]</code> Example: Router# show ip bgp 10.1.1.0	(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 12 <code>show ip bgp summary</code> Example: Router# 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 autonomous system number of 65536 displayed in the default asplain format.

```
RouterB# show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
  Advertised to update-groups:
    2
  65536
    192.168.1.2 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 autonomous system number 65536 for the BGP neighbor 192.168.1.2 of Router A in the figure above after this task has been configured on Router B:

```
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, page 50](#)

Troubleshooting Tips

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

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

Perform this task to modify the default output format for 4-byte autonomous system numbers from asplain format to asdot notation format. The **show ip bgp summary** command is used to display the changes in output format for the 4-byte autonomous system numbers.

This example requires Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXII, or a later release, to be running on the router.

SUMMARY STEPS

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

DETAILED STEPS

	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 ip bgp summary Example: Router# show ip bgp summary	Displays the status of all BGP connections.

	Command or Action	Purpose
Step 3	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 4	router bgp <i>autonomous-system-number</i> Example: <pre>Router(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 autonomous system number, 65538, is defined in asplain notation.
Step 5	bgp asnotation dot Example: <pre>Router(config-router)# bgp asnotation dot</pre>	Changes the default output format of BGP 4-byte autonomous system numbers from asplain (decimal values) to dot notation. <p>Note 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for show commands or the matching of regular expressions.</p>
Step 6	end Example: <pre>Router(config-router)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	clear ip bgp * Example: <pre>Router# 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 autonomous system 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: <pre>Router# show ip bgp summary</pre>	Displays the status of all BGP connections.
Step 9	show ip bgp regexp <i>regexp</i> Example: <pre>Router# show ip bgp regexp ^1\.0\$</pre>	Displays routes that match the autonomous system path regular expression. <ul style="list-style-type: none"> In this example, a regular expression to match a 4-byte autonomous system path is configured using asdot format.

Command or Action	Purpose
<p>Step 10 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p>Step 11 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(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 autonomous system number, 65538, is defined in asplain notation.
<p>Step 12 no bgp asnotation dot</p> <p>Example:</p> <pre>Router(config-router)# no bgp asnotation dot</pre>	<p>Resets the default output format of BGP 4-byte autonomous system numbers back to asplain (decimal values).</p> <p>Note 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for show commands or the matching of regular expressions.</p>
<p>Step 13 end</p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Exits router configuration mode and returns to privileged EXEC mode.
<p>Step 14 clear ip bgp *</p> <p>Example:</p> <pre>Router# clear ip bgp *</pre>	<p>Clears and resets all current BGP sessions.</p> <ul style="list-style-type: none"> In this example, a hard reset is performed to ensure that the 4-byte autonomous system number format change is reflected in all BGP sessions. <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 from the **show ip bgp summary** command shows the default asplain format of the 4-byte autonomous system numbers. Note the asplain format of the 4-byte autonomous system numbers, 65536 and 65550.

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

autonomous system numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 autonomous system numbers).

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  Statd
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 autonomous system paths is changed to asdot notation format. Although a 4-byte autonomous system number can be configured in a regular expression using either asplain format or asdot format, only 4-byte autonomous system numbers configured using the current default format are matched. In the first example below, the **show ip bgp regexp** command is configured with a 4-byte autonomous system number in asplain format. The match fails because the default format is currently asdot format and there is no output. In the second example using asdot format, the match passes and the information about the 4-byte autonomous system path is shown using the asdot notation.



Note

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

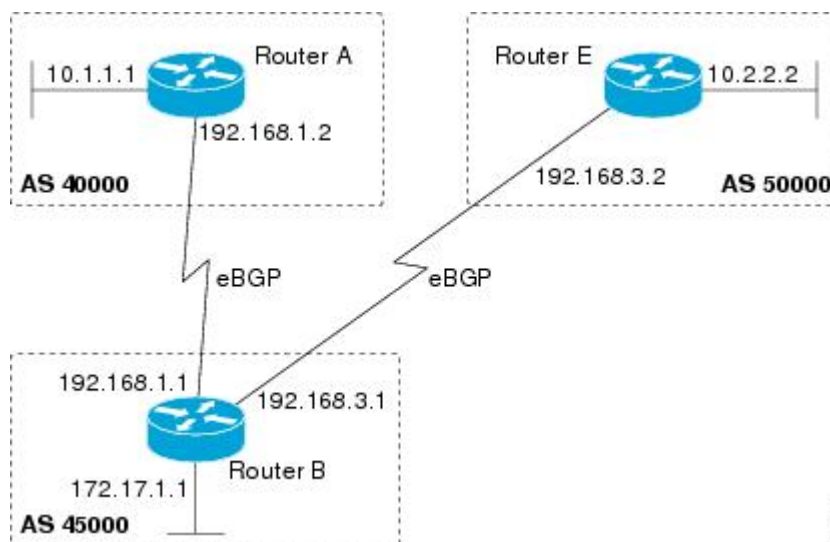
```
Router# show ip bgp regexp ^65536$
Router# show ip bgp regexp ^1\.0$
BGP table version is 2, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        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 8 BGP Topology for IPv4 VRF Address Family



Before you perform this task, perform the [Configuring a BGP Routing Process](#), page 40 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	<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>ip vrf <i>vrf-name</i></p> <p>Example:</p> <pre>Router(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 4	<p>rd <i>route-distinguisher</i></p> <p>Example:</p> <pre>Router(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 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 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>	<p>Exits VRF configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 7 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 8 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i></code></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.
<p>Step 9 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 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>
<p>Step 10 <code>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]</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 maximum-prefix 10000 warning-only</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. • 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.
<p>Step 11 <code>neighbor <i>ip-address</i> activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.</p>

Command or Action	Purpose
<p>Step 12 end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

- [Troubleshooting Tips, page 57](#)

Troubleshooting Tips

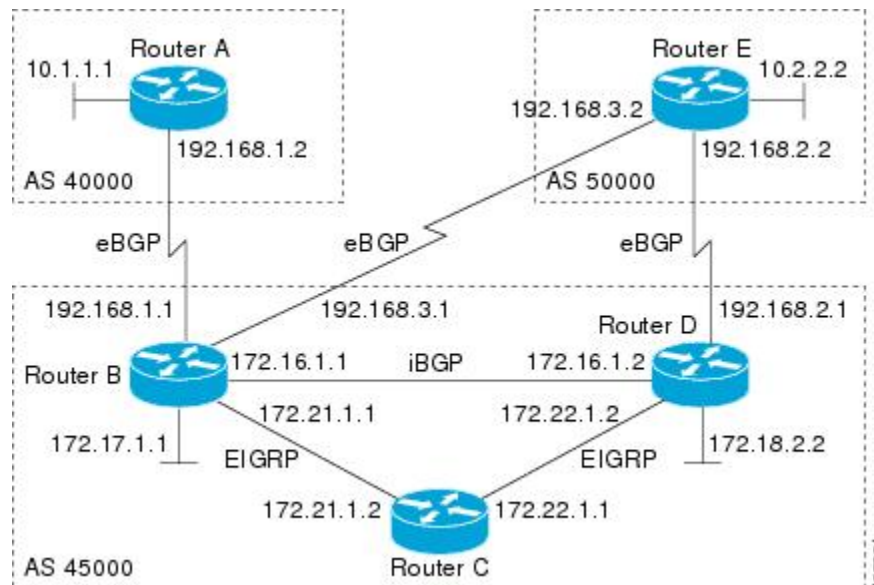
Use the **ping** command to verify basic network connectivity between the BGP routers, and use the **show ip vrf** command to verify that the VRF instance has been created.

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 routers.

Figure 9 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** *regex* | **dampened-routes** | **received prefix-filter**]

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.

	Command or Action	Purpose
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 for the specified routing process.
Step 4	<p>no bgp default ipv4-unicast</p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the IPv4 unicast address family for the BGP routing process.</p> <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</i> <i>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 of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} description <i>text</i></p> <p>Example:</p> <pre>Router(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>Router(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 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.

Command or Action	Purpose
<p>Step 8 <code>network network-number [mask network-mask] [route-map route-map-name]</code></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.
<p>Step 9 <code>neighbor {ip-address peer-group-name} activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 activate</pre>	<p>Enables the exchange of information with a BGP neighbor.</p>
<p>Step 10 <code>neighbor {ip-address peer-group-name} advertisement-interval seconds</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25</pre>	<p>(Optional) Sets the minimum interval between the sending of BGP routing updates.</p>
<p>Step 11 <code>neighbor {ip-address peer-group-name} default-originate[route-map map-name]</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 default-originate</pre>	<p>(Optional) Permits a BGP speaker--the local router--to send the default route 0.0.0.0 to a peer for use as a default route.</p>
<p>Step 12 <code>exit-address-family</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit-address- family</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 13 <code>neighbor {ip-address peer-group-name} shutdown</code></p> <p>Example:</p> <pre>Router(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>

Command or Action	Purpose
<p>Step 14 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>
<p>Step 15 <code>show ip bgp ipv4 multicast [command]</code></p> <p>Example:</p> <pre>Router# 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.
<p>Step 16 <code>show ip bgp neighbors [neighbor-address] [received-routes routes advertised-routes paths regexp dampened-routes received prefix-filter]</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors 192.168.3.2</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p>

Examples

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

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

Local Policy Denied Prefixes:
  Outbound      Inbound
  -----      -----
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](#), page 110.

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
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>no route-map map-name</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 4 <code>router eigrp autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config)# router eigrp 100</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 5 <code>no redistribute protocol [as-number]</code></p> <p>Example:</p> <pre>Router(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>
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>

Command or Action	Purpose
Step 7 <code>show running-config</code> Example: Router# <code>show running-config</code>	(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 routers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you may have to reset BGP connections for the configuration change to take effect.

- [Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing, page 64](#)
- [Resetting and Displaying Basic BGP Information, page 67](#)

Configuring Inbound Soft-Reconfiguration When Route Refresh Capability Is Missing

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

SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *autonomous-system-number*
- bgp log-neighbor-changes**
- bgp soft-reconfig-backup**
- neighbor** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number*
- neighbor** { *ip-address* | *peer-group-name* } **soft-reconfiguration**[inbound]
- neighbor** { *ip-address* | *peer-group-name* } **route-map** *map-name*{in | out}
- Repeat Steps 6 through 8 for every peer that is to be configured with soft-reconfiguration inbound.
- exit**
- route-map** *map-name* [permit | deny][sequence-number]
- set local-preference** *number-value*
- end**
- show ip bgp neighbors** [*neighbor-address*]
- 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 for the specified routing process.</p>
Step 4	<p>bgp log-neighbor-changes</p> <p>Example:</p> <pre>Router(config-router)# bgp log-neighbor-changes</pre>	<p>Enables logging of BGP neighbor resets.</p>
Step 5	<p>bgp soft-reconfig-backup</p> <p>Example:</p> <pre>Router(config-router)# bgp soft-reconfig-backup</pre>	<p>Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability.</p> <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	<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 of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</p>

Command or Action	Purpose
<p>Step 7 neighbor {<i>ip-address</i> <i>peer-group-name</i>} soft-reconfiguration[inbound]</p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound</pre>	<p>Configures the Cisco IOS software to start storing updates.</p> <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.
<p>Step 8 neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>map-name</i>{in out}</p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 route-map LOCAL in</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 LOCAL will be applied to incoming routes.
<p>Step 9 Repeat Steps 6 through 8 for every peer that is to be configured with soft-reconfiguration inbound.</p>	<p>--</p>
<p>Step 10 exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
<p>Step 11 route-map <i>map-name</i> [permit deny][sequence-number]</p> <p>Example:</p> <pre>Router(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.
<p>Step 12 set local-preference <i>number-value</i></p> <p>Example:</p> <pre>Router(config-route-map)# set local- preference 200</pre>	<p>Specifies a preference value for the autonomous system path.</p> <ul style="list-style-type: none"> In this example, the local preference value is set to 200.
<p>Step 13 end</p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route map configuration mode and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 14 <code>show ip bgp neighbors [neighbor-address]</code></p> <p>Example:</p> <pre>Router(config-router-af)# 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>
<p>Step 15 <code>show ip bgp [network] [network-mask]</code></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>

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
<p>Step 1 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.
<p>Step 2 clear ip bgp { * <i>autonomous-system-number</i> <i>neighbor-address</i> } [soft [in out]]</p> <p>Example:</p> <pre>Router# 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.
<p>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>]</p> <p>Example:</p> <pre>Router# 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.
<p>Step 4 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>Router# 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 router to BGP neighbor 192.168.3.2 on another router are displayed.

Command or Action	Purpose
Step 5 <code>show ip bgp paths</code> Example: Router# <code>show ip bgp paths</code>	Displays information about all the BGP paths in the database.
Step 6 <code>show ip bgp summary</code> Example: Router# <code>show ip bgp summary</code>	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, page 69](#)
- [Configuring Conditional Aggregate Routes Using BGP, page 71](#)
- [Suppressing and Unsuppressing Advertising Aggregated Routes Using BGP, page 72](#)
- [Suppressing Inactive Route Advertisement Using BGP, page 74](#)
- [Conditionally Advertising BGP Routes, page 76](#)

Redistributing a Static Aggregate Route into BGP

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

SUMMARY STEPS

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

DETAILED STEPS

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 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: <pre>Router(config)# ip route 172.0.0.0 255.0.0.0 null 0</pre>	Creates a static route.
Step 4 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 5 redistribute static Example: <pre>Router(config-router)# redistribute static</pre>	Redistributes routes into the BGP routing table.

Command or Action	Purpose
Step 6 <code>end</code> Example: <code>Router(config-router)# end</code>	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 AS-SET Information Generation" 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 <code>enable</code> Example: <code>Router> enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 <code>configure terminal</code> Example: <code>Router# configure terminal</code>	Enters global configuration mode.
Step 3 <code>router bgp <i>autonomous-system-number</i></code> Example: <code>Router(config)# router bgp 45000</code>	Enters router configuration mode for the specified routing process.

Command or Action	Purpose
<p>Step 4 <code>aggregate-address address mask [as-set]</code></p> <p>Example:</p> <pre>Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set</pre>	<p>Creates an aggregate entry in a BGP routing table.</p> <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>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>

Suppressing and Unsuppressing Advertising Aggregated Routes Using BGP

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

SUMMARY STEPS

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

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 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>
<p>Step 4 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 IPv4 multiprotocol BGP neighbor table of the local router.</p>
<p>Step 5 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>Router(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only</pre> <p>Example:</p> <pre>Router(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>

Command or Action	Purpose
<p>Step 6 <code>neighbor {ip-address peer-group-name}</code> <code>unsuppress-map map-name</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 unsuppress map1</pre>	<p>(Optional) Selectively advertises routes previously suppressed by the <code>aggregate-address</code> 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.
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>

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.

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

- enable**
- configure terminal**
- router bgp** *as-number*
- address-family** { *ipv4* [*mdt* | *multicast* | *unicast* [*vrf vrf-name*] | *vrf vrf-name*] | *vpn4* [*unicast*]}
- bgp suppress-inactive**
- end**
- show ip bgp rib-failure**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>as-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode and creates a BGP routing process.</p>
<p>Step 4 <code>address-family {ipv4 [<i>mdt</i> <i>multicast</i> <i>unicast</i> [<i>vrf vrf-name</i>] <i>vrf vrf-name</i>] <i>vpn4</i> [<i>unicast</i>]}</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Enter address family configuration mode to configure BGP peers to accept address family specific configurations.</p> <ul style="list-style-type: none"> The example creates an IPv4 unicast address family session.
<p>Step 5 <code>bgp suppress-inactive</code></p> <p>Example:</p> <pre>Router(config-router-af)# bgp suppress-inactive</pre>	<p>Suppresses BGP advertising of inactive routes.</p> <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.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
<p>Step 7 <code>show ip bgp rib-failure</code></p> <p>Example:</p> <pre>Router# show ip bgp rib-failure</pre>	<p>(Optional) Displays BGP routes that are not installed in the RIB.</p>

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

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

SUMMARY STEPS

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

DETAILED STEPS

	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> <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 of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.</p>
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>Router(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 router.</p> <ul style="list-style-type: none"> In this example, the prefix (172.17.0.0) matching the ACL in the advertise map (route map named map1) will be advertised to neighbor only when a prefix (192.168.50.0) matching the ACL in exist map (route-map "map2") is in the local BGP table.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 7 <code>route-map map-tag [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 8 <code>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...]}</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 9 <code>route-map map-tag [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 10 <code>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...]}</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 11 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 12 <code>access-list access-list-number {deny permit} source [source-wildcard] [log]</code></p> <p>Example:</p> <pre>Router(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.

Command or Action	Purpose
<p>Step 13 <code>access-list access-list-number {deny permit} source [source-wildcard] [log]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 14 <code>exit</code></p> <p>Example:</p> <pre>Router(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, page 79](#)
- [Conditionally Injecting BGP Routes, page 81](#)
- [Originating BGP Routes Using Backdoor Routes, page 85](#)

Advertising a Default Route Using BGP

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

SUMMARY STEPS

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

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>ip prefix-list list-name [seq seq-value] {deny network / length permit network / length} [ge ge-value] [le le-value]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 4 <code>route-map map-tag [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 5 <code>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...]}</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 6 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 7 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	Enters router configuration mode for the specified routing process.
<p>Step 8 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} default-originate[<i>route-map map-name</i>]</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.2 default-originate</pre>	(Optional) Permits a BGP speaker--the local router--to send the default route 0.0.0.0 to a peer for use as a default route.
<p>Step 9 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

- [Troubleshooting Tips, page 81](#)

Troubleshooting Tips

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

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

Conditionally Injecting BGP Routes

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp inject-map** *inject-map-name exist-map exist-map-name* [**copy-attributes**]
5. **exit**
6. **route-map** *map-tag* [**permit**| **deny**][*sequence-number*]
7. **match ip address** {*access-list-number* [*access-list-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. <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>bgp inject-map <i>inject-map-name</i> exist-map <i>exist-map-name</i> [copy-attributes]</p> <p>Example:</p> <pre>Router(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH</pre>	<p>Specifies the inject map and the exist map for conditional route injection.</p> <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>	<p>Exits router configuration mode and enters global configuration mode.</p>
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>	<p>Configures a route map and enters route map configuration mode.</p>
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>	<p>Exits route map configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 10 <code>route-map map-tag [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(config)# route-map ORIGINATE permit 10</pre>	<p>Configures a route map and enters route map configuration mode.</p>
<p>Step 11 <code>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...]}</code></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.
<p>Step 12 <code>set community {community-number [additive] [well-known-community] none}</code></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>
<p>Step 13 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 14 <code>ip prefix-list list-name [seq seq-value] {deny network / length permit network / length} [ge ge-value] [le le-value]</code></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>.
<p>Step 15 Repeat Step 14 for every prefix list to be created.</p>	<p>--</p>
<p>Step 16 <code>exit</code></p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>

Command or Action	Purpose
Step 17 <code>show ip bgp injected-paths</code> Example: Router# <code>show ip bgp injected-paths</code>	(Optional) Displays information about injected paths.

Examples

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

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

- [Troubleshooting Tips, page 85](#)

Troubleshooting Tips

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

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

Originating BGP Routes Using Backdoor Routes

Use this task to indicate to border routers which networks are reachable using a backdoor route. A backdoor network is treated the same as a local network except that it is not advertised. For more information see the [BGP Backdoor Routes, page 34](#).

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

SUMMARY STEPS

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

DETAILED STEPS

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</i> remote-as <i>autonomous-system-number</i> Example: <pre>Router(config-router)# neighbor 172.22.1.2 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 router. <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: <pre>Router(config-router)# network 172.21.1.0 backdoor</pre>	Indicates a network that is reachable through a backdoor route.

Command or Action	Purpose
Step 6 <code>end</code> Example: <code>Router(config-router)# end</code>	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	<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 40000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
Step 4	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Router(config-router)# neighbor fingroup peer-group</pre>	<p>Creates a BGP peer group.</p>
Step 5	<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 multiprotocol BGP neighbor table of the local router.</p>
Step 6	<p>neighbor <i>ip-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 peer-group fingroup</pre>	<p>Assigns the IP address of a BGP neighbor to a peer group.</p>

Command or Action	Purpose
<p>Step 7 <code>address-family ipv4 [unicast multicast vrf vrf-name]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 8 <code>neighbor peer-group-name activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor fingroup activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 address family with the local router.</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>
<p>Step 9 <code>neighbor ip-address peer-group peer-group-name</code></p> <p>Example:</p> <pre>Router(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>
<p>Step 10 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Configuring Peer Session Templates

The following tasks create and configure a peer session template:

- [Configuring a Basic Peer Session Template, page 89](#)
- [Configuring Peer Session Template Inheritance with the inherit peer-session Command, page 92](#)
- [Configuring Peer Session Template Inheritance with the neighbor inherit peer-session Command, page 94](#)

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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 101</pre>	Enters router configuration mode and creates a BGP routing process.
<p>Step 4 <code>template peer-session <i>session-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router)# template peer-session INTERNAL-BGP</pre>	Enters session-template configuration mode and creates a peer session template.
<p>Step 5 <code>remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router-stmp)# remote-as 202</pre>	<p>(Optional) Configures peering with a remote neighbor in the specified autonomous system.</p> <p>Note Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.</p>
<p>Step 6 <code>timers <i>keepalive-interval hold-time</i></code></p> <p>Example:</p> <pre>Router(config-router-stmp)# timers 30 300</pre>	<p>(Optional) Configures BGP keepalive and hold timers.</p> <ul style="list-style-type: none"> The hold time must be at least twice the keepalive time. <p>Note Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.</p>
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Exits session-template configuration mode and returns to privileged EXEC mode.
<p>Step 8 <code>show ip bgp template peer-session [<i>session-template-name</i>]</code></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 <i>session-template-name</i> argument. This command also supports all standard output modifiers.

- [What to Do Next, page 91](#)

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 <code>enable</code> Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>router bgp</code> <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 101</pre>	Enters router configuration mode and creates a BGP routing process.

Command or Action	Purpose
<p>Step 4 <code>template peer-session <i>session-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router)# template peer-session CORE1</pre>	<p>Enter session-template configuration mode and creates a peer session template.</p>
<p>Step 5 <code>description <i>text-string</i></code></p> <p>Example:</p> <pre>Router(config-router-stmp)# description CORE-123</pre>	<p>(Optional) Configures a description.</p> <ul style="list-style-type: none"> The text string can be up to 80 characters. <p>Note Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.</p>
<p>Step 6 <code>update-source <i>interface-type interface-number</i></code></p> <p>Example:</p> <pre>Router(config-router-stmp)# update-source loopback 1</pre>	<p>(Optional) Configures a router to select a specific source or interface to receive routing table updates.</p> <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. <p>Note Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.</p>
<p>Step 7 <code>inherit peer-session <i>session-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router-stmp)# inherit peer-session INTERNAL-BGP</pre>	<p>Configures this peer session template to inherit the configuration of another peer session template.</p> <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.
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits session-template configuration mode and enters privileged EXEC mode.</p>
<p>Step 9 <code>show ip bgp template peer-session [<i>session-template-name</i>]</code></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, page 94](#)

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 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.

Command or Action	Purpose
<p>Step 4 <code>neighbor ip-address remote-as autonomous-system-number</code></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.
<p>Step 5 <code>neighbor ip-address inherit peer-session session-template-name</code></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.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>
<p>Step 7 <code>show ip bgp template peer-session [session-template-name]</code></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, page 95](#)

What to Do Next

To create a peer policy template, go to the [Configuring Peer Policy Templates, page 95](#).

Configuring Peer Policy Templates

The following tasks create and configure a peer policy template:

- [Configuring Basic Peer Policy Templates, page 96](#)
- [Configuring Peer Policy Template Inheritance with the inherit peer-policy Command, page 98](#)
- [Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command, page 100](#)

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 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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode and creates a BGP routing process.</p>
<p>Step 4 <code>template peer-policy <i>policy-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router)# template peer-policy GLOBAL</pre>	<p>Enters policy-template configuration mode and creates a peer policy template.</p>
<p>Step 5 <code>maximum-prefix <i>prefix-limit</i> [<i>threshold</i>] [<i>restart restart-interval</i> <i>warning-only</i>]</code></p> <p>Example:</p> <pre>Router(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, page 38.</p>
<p>Step 6 <code>weight <i>weight-value</i></code></p> <p>Example:</p> <pre>Router(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, page 38.</p>
<p>Step 7 <code>prefix-list <i>prefix-list-name</i> {in out}</code></p> <p>Example:</p> <pre>Router(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, page 38.</p>
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-ptmp)# end</pre>	<p>Exits policy-template configuration mode and returns to privileged EXEC mode.</p>

- [What to Do Next, page 97](#)

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template. For more details about peer policy inheritance see the

"Configuring Peer Policy Template Inheritance with the inherit peer-policy Command" section or the "Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command" section.

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

This task configures peer policy template inheritance using the **inherit peer-policy** command. It creates and 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 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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode and creates a BGP routing process.</p>
<p>Step 4 <code>template peer-policy <i>policy-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router)# template peer-policy NETWORK1</pre>	<p>Enter policy-template configuration mode and creates a peer policy template.</p>
<p>Step 5 <code>route-map <i>map-name</i> {in out}</code></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, page 38.</p>
<p>Step 6 <code>inherit peer-policy <i>policy-template-name</i> <i>sequence-number</i></code></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.
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-ptmp)# end</pre>	<p>Exits policy-template configuration mode and returns to privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 8 <code>show ip bgp template peer-policy [policy-template-name[detail]]</code></p> <p>Example:</p> <pre>Router# show ip bgp template peer-policy NETWORK1 detail</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 <code>policy-template-name</code> 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
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 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>
<p>Step 4 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>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 6 to work. If a peering is not configured, the specified neighbor in Step 6 will not accept the session template.
<p>Step 5 address-family ipv4 [multicast unicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address- family ipv4 unicast</pre>	<p>Enters address family configuration mode to configure a neighbor to accept address family-specific command configurations.</p>

Command or Action	Purpose
<p>Step 6 <code>neighbor ip-address inherit peer-policy policy-template-name</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL</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"> 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.
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
<p>Step 8 <code>show ip bgp neighbors [ip-address[policy [detail]]]</code></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 policies:
  prefix-list NO-MARKETING in
  route-map ROUTE in
  weight 300
  maximum-prefix 10000
```

Monitoring and Maintaining BGP Dynamic Update Groups

Use this task to clear and display information about the processing of dynamic BGP update groups. The performance of BGP update message generation is improved with the use of BGP update groups. With the configuration of the BGP peer templates and the support of the dynamic BGP update groups, the network

operator no longer needs to configure peer groups in BGP and can benefit from improved configuration flexibility and system performance. For more information about using BGP peer templates, see "Configuring Peer Session Templates" and "Configuring Peer Policy Templates".

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: Router> 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: Router# 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: Router# 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: Router# show ip bgp update-group	Displays information about BGP update groups.

- [Troubleshooting Tips, page 103](#)

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, page 104](#)
- [Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers, page 105](#)
- [Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number, page 107](#)
- [Example NLRI to AFI Configuration, page 108](#)
- [Examples Removing BGP Configuration Commands Using a Redistribution Example, page 110](#)
- [Examples BGP Soft Reset, page 111](#)
- [Example Resetting BGP Peers Using 4-Byte Autonomous System Numbers, page 111](#)
- [Example Resetting and Displaying Basic BGP Information, page 112](#)
- [Examples Aggregating Prefixes Using BGP, page 113](#)
- [Example Configuring a BGP Peer Group, page 114](#)
- [Example Configuring Peer Session Templates, page 115](#)
- [Example Configuring Peer Policy Templates, page 115](#)
- [Examples Monitoring and Maintaining BGP Dynamic Update Peer-Groups, page 116](#)

Example Configuring a BGP Process and Customizing Peers

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

Router B

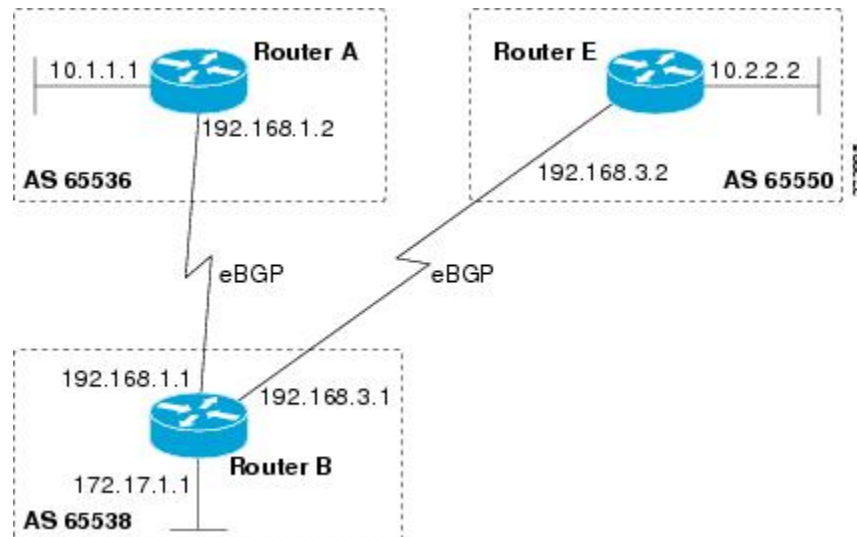
```
router bgp 45000
  bgp router-id 172.17.1.99
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  timers bgp 70 120
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.3.2 remote-as 50000
  neighbor 192.168.3.2 description finance
  !
  address-family ipv4
    neighbor 192.168.1.2 activate
    neighbor 192.168.3.2 activate
    no auto-summary
    no synchronization
    network 172.17.1.0 mask 255.255.255.0
  exit-address-family
  !
  address-family ipv4 multicast
    neighbor 192.168.3.2 activate
    neighbor 192.168.3.2 advertisement-interval 25
    no auto-summary
    no synchronization
    network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

Examples: Configuring a BGP Routing Process and Peers 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)SXI1, 15.1(1)SG, 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)SXI1, 15.1(1)SG, and later releases and shows the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using asplain notation. IPv4 unicast routes are exchanged with all peers.

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



Router A

```

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

```

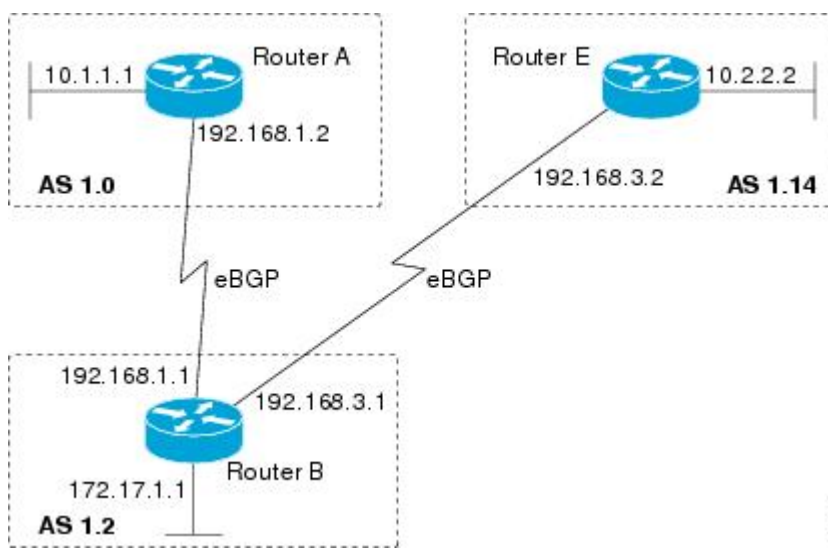
router bgp 65550
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family

```

Asdot 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 the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

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



30 903.1

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
```

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 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
```

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

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

CLI to Enable BGP Route Redistribution Into EIGRP

```

route-map bgp-to-eigrp permit 10
  match tag 50000
  set tag 65000
  exit
router bgp 45000
  bgp log-neighbor-changes
  address-family ipv4
    neighbor 172.16.1.2 remote-as 45000
    neighbor 172.21.1.2 remote-as 45000
    neighbor 192.168.1.2 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.

Dynamic Inbound Soft Reset Example

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

```
clear ip bgp 192.168.1.1 soft in
```

Inbound Soft Reset Using Stored Information Example

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

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

The following example clears the session with the neighbor 192.168.1.1:

```
clear ip bgp 192.168.1.1 soft in
```

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. This example requires Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXII, or a later release to be running on the router. The initial state of the BGP routing table is shown using the **show ip bgp** command, and peers in 4-byte autonomous systems 65536 and 65550 are displayed.

```
RouterB# show ip bgp

BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 10.1.1.0/24    192.168.1.2         0             0 65536 i
*> 10.2.2.0/24    192.168.3.2         0             0 65550 i
*> 172.17.1.0/24  0.0.0.0             0             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             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, starting in global configuration mode, configures BGP to not advertise inactive routes:

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

Router(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:

```
Router(config)# ip vrf RED

Router(config-vrf)# rd 50000:10
Router(config-vrf)# maximum routes 1000 10

Router(config-vrf)# exit
Router(config)# router bgp 50000
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# bgp suppress-inactive

Router(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
```

Example Configuring Peer Policy Templates

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

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

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

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

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

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

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

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

Examples Monitoring and Maintaining BGP Dynamic Update Peer-Groups

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

clear ip bgp update-group Example

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

```
Router#
 clear ip bgp update-group 10.0.0.1
```

debug ip bgp groups Example

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

```
Router# debug ip bgp groups
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.5 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 fl0
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 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.8 fl0
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 NLRI 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 NLRI 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 Commands List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS IPv6 Command Reference</i>
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	"Cisco BGP Overview" module
Multiprotocol Label Switching (MPLS) and BGP configuration example using the IPv4 VRF address family	"Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels" module
Basic MPLS VPN and BGP configuration example	"Configuring MPLS Layer 3 VPNs" module

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	A Border Gateway Protocol 4 (BGP-4)
RFC 4893	BGP Support for Four-octet AS Number Space
RFC 5396	Textual Representation of Autonomous system (AS) Numbers
RFC 5398	Autonomous System (AS) Number Reservation for Documentation Use

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11 Feature Information for Configuring a Basic BGP Network

Feature Name	Releases	Feature Configuration Information
BGP Version 4	Cisco IOS XE 3.1.0SG	BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS).

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.

Feature Name	Releases	Feature Configuration Information
BGP Neighbor Policy	12.2(33)SB 12.2(33)SRB 12.4(11)T Cisco IOS XE 3.1.0SG 15.0(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 by this feature: show ip bgp neighbors, show ip bgp template peer-policy.</p>

Feature Name	Releases	Feature Configuration Information
BGP Support for 4-Byte ASN	12.0(32)S 12.0(32)SY8 12.0(33)S3 12.2(33)SRE 12.2(33)XNE 12.2(33)SX11 12.4(24)T 15.0(1)S Cisco IOS XE 3.1.0SG	<p>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.</p> <p>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 <code>asplain</code> as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the <code>asplain</code> format and the <code>asdot</code> format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to <code>asdot</code> format, use the <code>bgp asnotation dot</code> command.</p> <p>In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses <code>asdot</code> as the only configuration format, regular expression match, and output display, with no <code>asplain</code> support.</p> <p>The following commands were introduced or modified by this feature: <code>bgp asnotation dot</code>, <code>bgp confederation identifier</code>, <code>bgp confederation peers</code>, all <code>clear ip bgp</code> commands that configure an autonomous system number, <code>ip as-path access-list</code>, <code>ip extcommunity-list</code>, <code>match source-protocol</code>, <code>neighbor local-</code></p>

Feature Name	Releases	Feature Configuration Information
Suppress BGP Advertisement for Inactive Routes	12.2(25)S 12.2(33)SXH 15.0(1)M 15.0(1)S	<p>as, neighbor remote-as, neighbor soo, redistribute (IP), router bgp, route-target, set as-path, set extcommunity, set origin, soo, all show ip bgp commands that display an autonomous system number, and show ip extcommunity-list.</p> <p>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.</p>

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: 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)

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



Connecting to a Service Provider Using External BGP

This module describes configuration tasks that will enable your Border Gateway Protocol (BGP) network to access peer devices in external networks such as those from Internet service providers (ISPs). BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. External BGP (eBGP) peering sessions are configured to allow peers from different autonomous systems to exchange routing updates. Tasks to help manage the traffic that is flowing inbound and outbound are described, as are tasks to configure BGP policies to filter the traffic. Multihoming techniques that provide redundancy for connections to a service provider are also described.

- [Finding Feature Information, page 125](#)
- [Prerequisites for Connecting to a Service Provider Using External BGP, page 125](#)
- [Restrictions for Connecting to a Service Provider Using External BGP, page 126](#)
- [Information About Connecting to a Service Provider Using External BGP, page 126](#)
- [How to Connect to a Service Provider Using External BGP, page 138](#)
- [Configuration Examples for Connecting to a Service Provider Using External BGP, page 196](#)
- [Where to Go Next, page 207](#)
- [Additional References, page 208](#)
- [Feature Information for Connecting to a Service Provider Using External BGP, page 209](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Connecting to a Service Provider Using External BGP

- Before connecting to a service provider you need to understand how to configure the basic BGP process and peers. See the "Cisco BGP Overview" and "Configuring a Basic BGP Network" modules for more details.

- The tasks and concepts in this chapter will help you configure BGP features that would be useful if you are connecting your network to a service provider. For each connection to the Internet, you must have an assigned autonomous system number from the Internet Assigned Numbers Authority (IANA).

Restrictions for Connecting to a Service Provider Using External BGP

- A router that runs Cisco IOS 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, page 126](#)
- [BGP Autonomous System Number Formats, page 128](#)
- [BGP Attributes, page 130](#)
- [Multihoming, page 131](#)
- [MED Attribute, page 132](#)
- [Transit Versus Nontransit Traffic, page 132](#)
- [BGP Policy Configuration, page 132](#)
- [BGP Prefix-Based Outbound Route Filtering, page 133](#)
- [BGP Communities, page 134](#)
- [Extended Communities, page 134](#)
- [Extended Community Lists, page 135](#)
- [Administrative Distance, page 136](#)
- [BGP Route Map Policy Lists, page 136](#)
- [BGP Route Map with a Continue Clause, page 136](#)

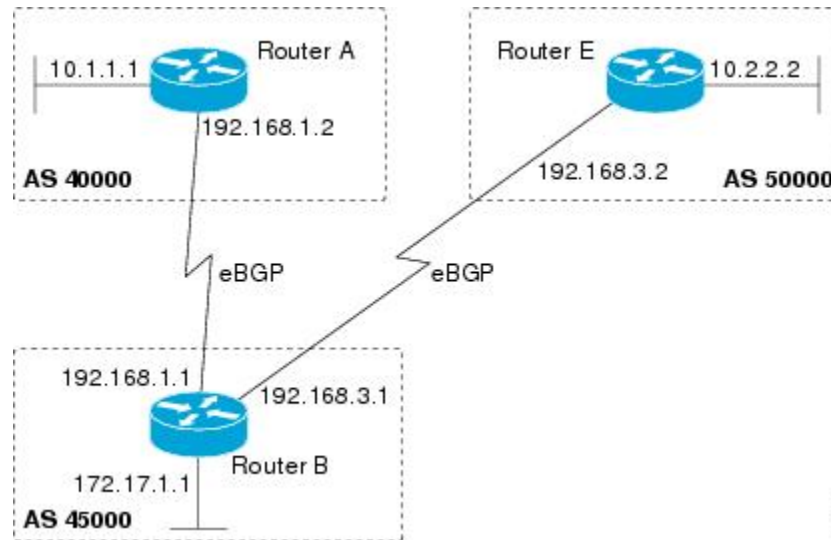
External BGP Peering

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 12 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 12 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)SXII, 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 13 *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 14 *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)SXI1, 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 IOS software is to treat routes without the MED attribute as having a MED of 0, making the route that lacks the MED variable the most preferred. To configure the router to conform to the IETF standard, use the **bgp bestpath med missing-as-worst** router configuration command.

Next_Hop

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

Origin

This attribute indicates how the route was included in a BGP routing table. In Cisco IOS 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.

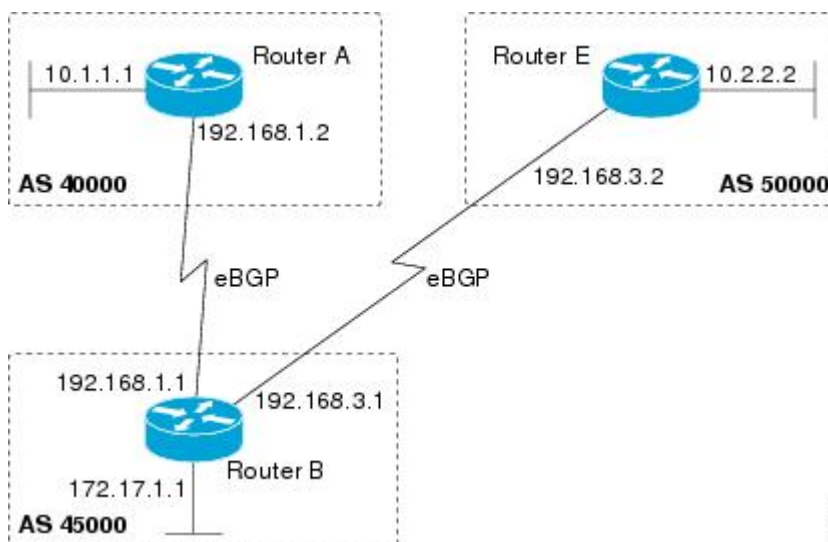
Multihoming

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 13 **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 15 *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 Prefix-Based Outbound Route Filtering

BGP prefix-based outbound route filtering uses the BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers. Configuring BGP ORF can help reduce the amount of system resources required for generating and processing routing updates by filtering out unwanted routing updates at the source. For example, BGP ORF can be used to reduce the amount of processing required on a router that is not accepting full routes from a service provider network.

The BGP prefix-based outbound route filtering is enabled through the advertisement of ORF capabilities to peer routers. The advertisement of the ORF capability indicates that a BGP peer will accept a prefix list from a neighbor and apply the prefix list to locally configured ORFs (if any exist). When this capability is

enabled, the BGP speaker can install the inbound prefix list filter to the remote peer as an outbound filter, which reduces unwanted routing updates.

The BGP prefix-based outbound route filtering can be configured with send or receive ORF capabilities. The local peer advertises the ORF capability in send mode. The remote peer receives the ORF capability in receive mode and applies the filter as an outbound policy. The local and remote peers exchange updates to maintain the ORF on each router. Updates are exchanged between peer routers by address family depending on the ORF prefix list capability that is advertised. The remote peer starts sending updates to the local peer after a route refresh has been requested with the **clear ip bgp in prefix-filter** command or after an ORF prefix list with immediate status is processed. The BGP peer will continue to apply the inbound prefix list to received updates after the local peer pushes the inbound prefix list to the remote peer.

BGP Communities

BGP communities are used to group routes (also referred to as color routes) that share common properties, regardless of network, autonomous system, or any physical boundaries. In large networks applying a common routing policy through prefix-lists or access-lists requires individual peer statements on each networking device. Using the BGP community attribute BGP speakers, with common routing policies, can implement inbound or outbound route filters based on the community tag rather than consult large lists of individual permit or deny statements.

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

The community attribute is optional, which means that it will not be passed on by networking devices that do not understand communities. Networking devices that understand communities must be configured to handle the communities or they will be discarded.

There are four predefined communities:

- no-export--Do not advertise to external BGP peers.
- no-advertise--Do not advertise this route to any peer.
- internet--Advertise this route to the Internet community; all BGP-speaking networking devices belong to it.
- local-as--Do not send outside the local autonomous system.

In Cisco IOS Release 12.2(8)T, BGP named community lists were introduced. BGP named community lists allow meaningful names to be assigned to community lists with no limit on 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 the rules of numbered communities apply to named community lists except that there is no limitation on the number of named community lists that can be configured.



Note

Both standard and expanded community lists have a limitation of 100 community groups that can be configured within each type of list. A named community list does not have this limitation.

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 list functions like a macro when it is configured in a route map and has the following capabilities and characteristics:

- When a policy list is referenced within a route map, all the match statements within the policy list are evaluated and processed.
- Two or more policy lists can be configured with a route map. Policy lists can be configured within a route map to be evaluated with AND or OR semantics.
- Policy lists can coexist with any other preexisting match and set statements that are configured within the same route map but outside of the policy lists.
- When multiple policy lists perform matching within a route map entry, all policy lists match on the incoming attribute only.

Policy lists support only match clauses and do not support set clauses. Policy lists can be configured for all applications of route maps, including redistribution, and can also coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists.

**Note**

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

BGP Route Map with a Continue Clause

In Cisco IOS Release 12.3(2)T, 12.0(24)S, 12.2(33)SRB, and later releases, the continue clause was introduced into BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduced the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. Before the continue clause was introduced, route map configuration was linear and did not allow any control over the flow of a route map.

In Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases, support for continue clauses for outbound route maps was introduced.

- [Route Map Operation Without Continue Clauses, page 137](#)
- [Route Map Operation with Continue Clauses, page 137](#)
- [Match Operations with Continue Clauses, page 137](#)

- [Set Operations with Continue Clauses, page 137](#)

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map "falls through" and evaluates the next sequence number of the route map until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order starting with the lowest sequence number and ending with the highest sequence number. If the route map contains only set clauses, the set clauses will be executed automatically, and the route map will not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map will continue to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (or jump to) a specific route map entry by specifying the sequence number, or if a sequence number is not specified, the continue clause will go to the next sequence number. This behavior is called an "implied continue." If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

Match Operations with Continue Clauses

If a match clause does not exist in the route map entry but a continue clause does, the continue clause will be automatically executed and go to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map entry contains a continue clause, the route map will execute the continue clause if a successful match occurs. If a continue clause does not exist in the next route map entry, the route map will be evaluated normally. If a continue clause exists in the next route map entry but a match does not occur, the route map will not continue and will "fall through" to the next sequence number if one exists.

Set Operations with Continue Clauses

Set clauses are saved during the match clause evaluation process and executed after the route-map evaluation is completed. The set clauses are evaluated and executed in the order in which they were configured. Set clauses are executed only after a successful match occurs, unless the route map does not contain a match clause. The continue statement proceeds to the specified route map entry only after configured set actions are performed. If a set action occurs in the first route map and then the same set action occurs again, with a different value, in a subsequent route map entry, the last set action may override any previous set actions that were configured with the same **set** command unless the **set** command permits more than one value. For example, the **set as-path prepend** command permits more than one autonomous system number to be configured.



Note

A continue clause can be executed, without a successful match, if a route map entry does not contain a match clause.

**Note**

Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. For an example, see "Filtering Traffic Using Continue Clauses in a BGP Route Map Examples".

- [Restrictions, page 138](#)

Restrictions

- Continue clauses for outbound route maps are supported only in Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases.
- Continue clauses can go only to a higher route map entry (a route map entry with a higher sequence number) and cannot go to a lower route map entry.

How to Connect to a Service Provider Using External BGP

- [Influencing Inbound Path Selection, page 138](#)
- [Influencing Outbound Path Selection, page 146](#)
- [Configuring BGP Peering with ISPs, page 153](#)
- [Configuring BGP Policies, page 169](#)

Influencing Inbound Path Selection

BGP can be used to influence the choice of paths in another autonomous system. There may be several reasons for wanting BGP to choose a path that is not the obvious best route, for example, to avoid some types of transit traffic passing through an autonomous system or perhaps to avoid a very slow or congested link. BGP can influence inbound path selection using one of the following BGP attributes:

- AS-path
- MED

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

- [Influencing Inbound Path Selection by Modifying the AS-path Attribute, page 138](#)
- [Influencing Inbound Path Selection by Setting the MED Attribute, page 143](#)

Influencing Inbound Path Selection by Modifying the AS-path Attribute

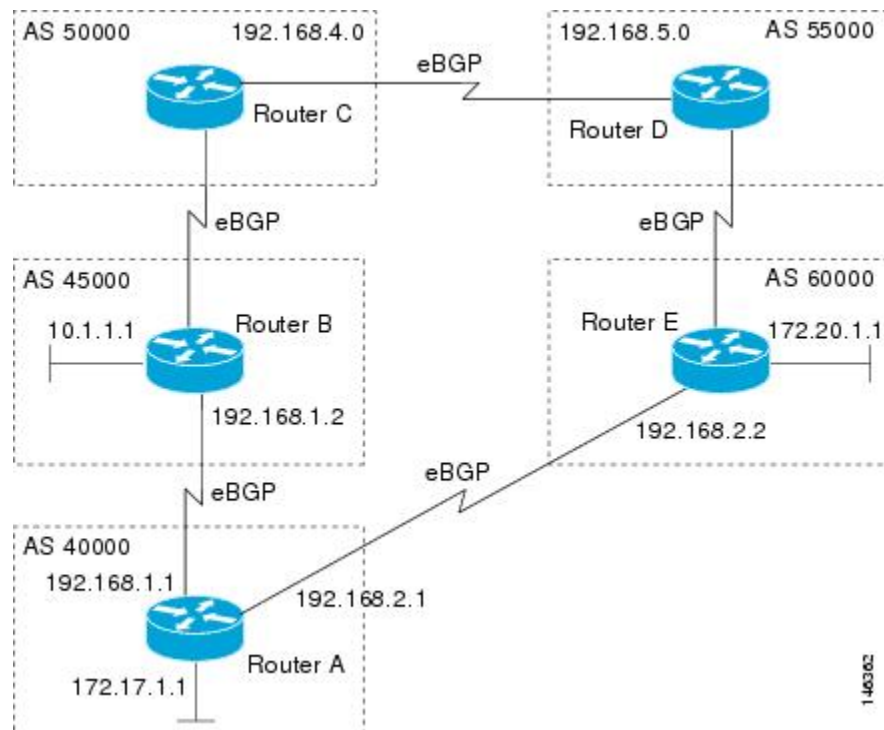
Perform this task to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. The configuration is performed at Router A in the figure below. For a configuration example of this task using 4-byte autonomous system numbers in asplain format, see [Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example, page 197](#).

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 14 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* } **route-map** *map-name*{**in** | **out** }
8. **neighbor** { *ip-address* | *peer-group-name* } **activate**
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	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	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.1.2 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 BGP peer on Router B at 192.168.1.2 is added to the IPv4 multiprotocol BGP neighbor table and will receive BGP updates.

	Command or Action	Purpose
Step 5	<p>address-family <i>ipv4</i> [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>] [route-map <i>route-map-name</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>} route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 route-map PREPEND 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 PREPEND is applied to outbound routes to Router B.
Step 8	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 activate</pre>	<p>Enables address exchange for address family IPv4 unicast for the BGP neighbor at 192.168.1.2 on Router B.</p>
Step 9	<p>exit-address-family</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
Step 10	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 11 <code>route-map map-name [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(config)# route-map PREPEND 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 PREPEND is created with a permit clause.
<p>Step 12 <code>set as-path {tag prepend as-path-string}</code></p> <p>Example:</p> <pre>Router(config-route-map)# set as-path prepend 40000 40000</pre>	<p>Modifies an autonomous system path for BGP routes.</p> <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.
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route map configuration mode and returns to privileged EXEC mode.</p>
<p>Step 14 <code>show running-config</code></p> <p>Example:</p> <pre>Router# show running-config</pre>	<p>Displays the running configuration file.</p>

Examples

Router A

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

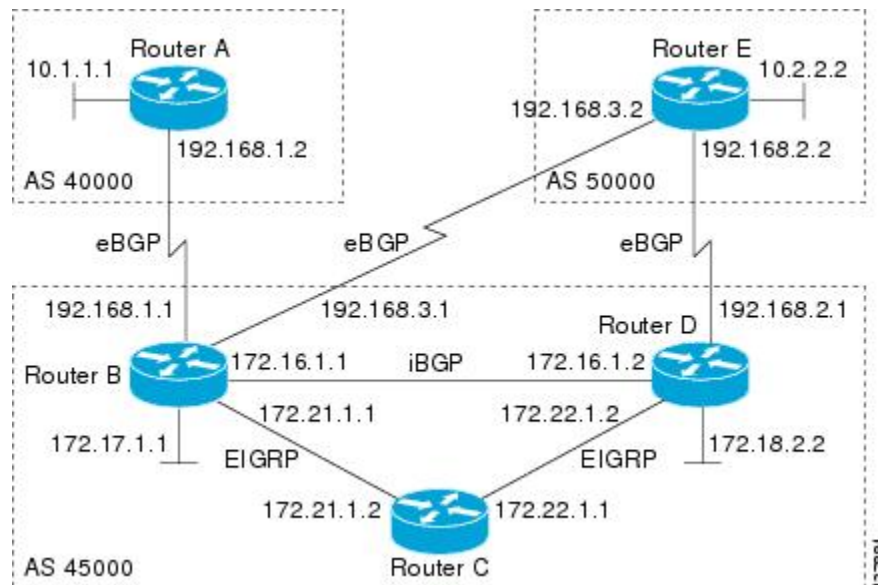
```
Router# show running-config
.
.
.
router bgp 40000
 neighbor 192.168.1.2 remote-as 45000
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-map PREPEND out
 no auto-summary
 no synchronization
 network 172.17.1.0 mask 255.255.255.0
 exit-address-family
 !
 route-map 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 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 15 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	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	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.3.2 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.

	Command or Action	Purpose
Step 5	<p>address-family <i>ipv4</i> [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>] [route-map <i>route-map-name</i>]</p> <p>Example:</p> <pre>Router(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.
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>map-name</i>{in out}</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.2 route-map MED 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 MED is applied to outbound routes to the BGP peer at Router E.
Step 8	<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 9	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
Step 10	<p>route-map <i>map-name</i> [permit deny][<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map MED 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 MED is created.

Command or Action	Purpose
Step 11 <code>set metric value</code> Example: <pre>Router(config-route-map)# set metric 50</pre>	Sets the MED metric value.
Step 12 <code>end</code> Example: <pre>Router(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 <code>show ip bgp [network] [network-mask]</code> Example: <pre>Router# 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.

```
Router# show ip bgp 172.16.1.0
BGP routing table entry for 172.16.1.0/24, version 10
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
    1
  45000
    192.168.2.1 from 192.168.2.1 (192.168.2.1)
      Origin IGP, metric 120, localpref 100, valid, external
  45000
    192.168.3.1 from 192.168.3.1 (172.17.1.99)
      Origin IGP, metric 50, localpref 100, valid, external, best
```

Influencing Outbound Path Selection

BGP can be used to influence the choice of paths for outbound traffic from the local autonomous system. This section contains two methods that BGP can use to influence outbound path selection:

- Using the `Local_Pref` attribute
- Using the BGP outbound route filter (ORF) capability

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

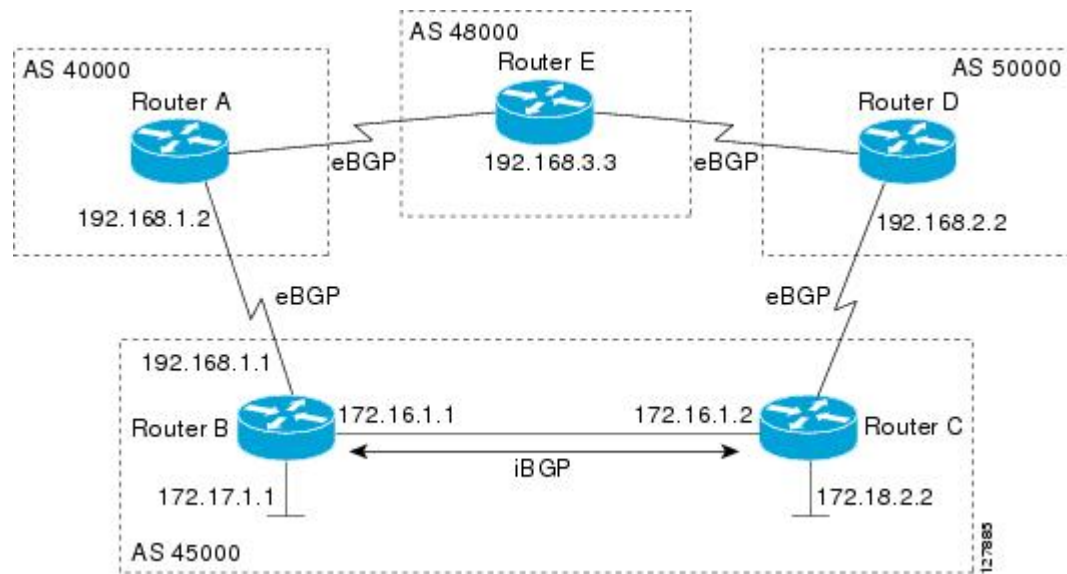
- [Influencing Outbound Path Selection Using the Local_Pref Attribute, page 147](#)
- [Filtering Outbound BGP Route Prefixes, page 150](#)

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 16 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.
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	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.1.2 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.

	Command or Action	Purpose
Step 5	bgp default local-preference <i>value</i> Example: <pre>Router(config-router)# bgp default local-preference 150</pre>	Changes the default local preference value. <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>	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>] [route-map <i>route-map-name</i>] Example: <pre>Router(config-router-af)# network 172.17.1.0 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.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> } 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.	--

Command or Action	Purpose
Step 11 <code>show ip bgp [network] [network-mask]</code> 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.

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

- enable**
- configure terminal**
- ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length* | **permit** *network / length*}[**ge** *ge-value*][**le** *le-value*]
- router bgp** *autonomous-system-number*
- neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
- neighbor** *ip-address* **ebgp-multihop** [*hop-count*]
- address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*]
- neighbor** *ip-address* **capability orf prefix-list** [**send** | **receive** | **both**]
- neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**}
- end**
- clear ip bgp** {*ip-address* | *} **in prefix-filter**

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>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 FILTER seq 10 permit 192.168.1.0/24</pre>	<p>Creates a prefix list for prefix-based outbound route filtering.</p> <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	<p>router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 100</pre>	<p>Enters router configuration mode, and creates a BGP routing process.</p>
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>

Command or Action	Purpose
<p>Step 7 <code>address-family ipv4 [unicast multicast] vrf vrf-name]</code></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>
<p>Step 8 <code>neighbor ip-address capability orf prefix-list [send receive both]</code></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.
<p>Step 9 <code>neighbor {ip-address peer-group-name} prefix-list prefix-list-name {in out}</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in</pre>	<p>Applies an inbound prefix-list filter to prevent distribution of BGP neighbor information.</p> <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.
<p>Step 10 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode, and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 11 <code>clear ip bgp {ip-address *} in prefix-filter</code></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 <code>clear ip bgp</code> 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, page 153](#)
- [Multihoming with a Single ISP, page 157](#)
- [Configuring Multihoming to Receive the Full Internet Routing Table, page 165](#)

Configuring Multihoming with Two ISPs

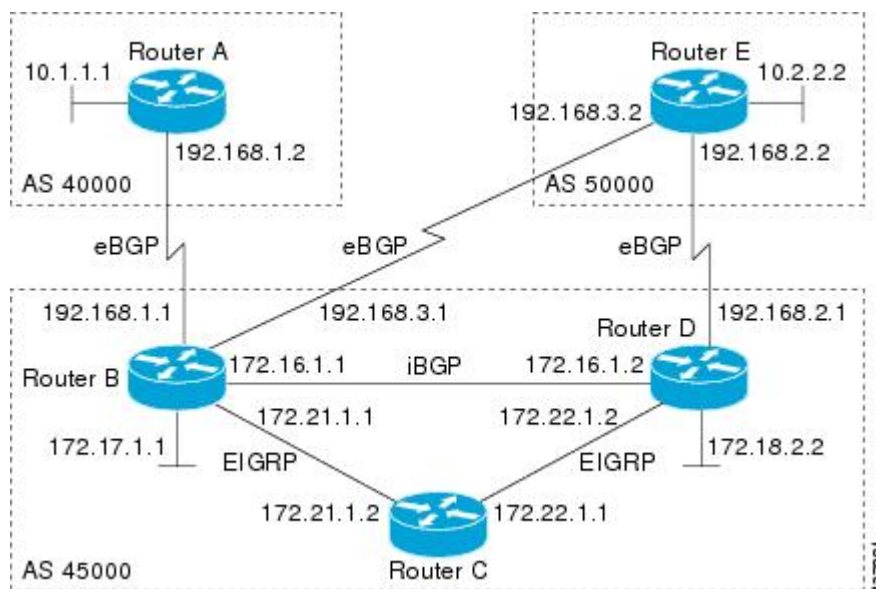
Perform this task to configure your network to access two ISPs, where one ISP is the preferred route and the second ISP is a backup route. In the figure below Router B in autonomous system 45000 has BGP peers in two ISPs, autonomous system 40000 and autonomous system 50000. Using this task, Router B 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 17 *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.

Command or Action	Purpose
<p>Step 7 <code>neighbor {ip-address peer-group-name} weight number</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 weight 150</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 192.168.1.2 is set to 150.
<p>Step 8 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 9 <code>neighbor {ip-address peer-group-name} remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(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>
<p>Step 10 <code>address-family ipv4 [unicast multicast] vrf vrf-name]</code></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>
<p>Step 11 <code>neighbor {ip-address peer-group-name} weight number</code></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.
<p>Step 12 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 13 <code>clear ip bgp</code> [*] <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>
<p>Step 14 <code>show ip bgp</code> [<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 192.168.3.2 (Router E in the figure above) has the highest weight attribute and will be the preferred route to network 172.17.1.0.

```
BGP table version is 8, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network        Next Hop           Metric LocPrf Weight Path
*> 10.1.1.0/24  192.168.1.2             0         100 40000 i
*> 10.2.2.0/24  192.168.3.2             0         150 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: 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 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: Router(config-router)# neighbor 192.168.2.1 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 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: Router(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	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] Example: Router(config-router-af)# network 10.2.2.0 mask 255.255.255.0	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
<p>Step 7 <code>neighbor {ip-address peer-group-name}</code> <code>route-map map-name{in out}</code></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>
<p>Step 8 Repeat Step 7 to apply another route map to the neighbor specified in Step 7.</p>	<p>--</p>
<p>Step 9 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 10 <code>neighbor {ip-address peer-group-name}</code> <code>remote-as autonomous-system-number</code></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.
<p>Step 11 <code>address-family ipv4 [unicast multicast vrf vrf-name]</code></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>

Command or Action	Purpose
<p>Step 12 neighbor {<i>ip-address</i> <i>peer-group-name</i>} 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>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.3.1 route-map SETMETRIC2 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 SETMETRIC2 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>
<p>Step 13 Repeat Step 10 to apply another route map to the neighbor specified in Step 10.</p>	<p>--</p>
<p>Step 14 exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 15 exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 16 <code>ip route prefix mask {ip-address interface-type interface-number[ip-address]} [distance] [name][permanent] track number[tag tag]</code></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>
<p>Step 17 Repeat Step 14 to establish another static route.</p>	<p>--</p>
<p>Step 18 <code>route-map map-name [permit deny] [sequence-number]</code></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.
<p>Step 19 <code>set metric value</code></p> <p>Example:</p> <pre>Router(config-route-map)# set metric 100</pre>	<p>Sets the MED metric value.</p>
<p>Step 20 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 21 <code>route-map map-name [permit deny]</code> <code>[sequence-number]</code></p> <p>Example:</p> <pre>Router(config)# route-map SETMETRIC2 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 SETMETRIC2 is created.
<p>Step 22 <code>set metric value</code></p> <p>Example:</p> <pre>Router(config-route-map)# set metric 50</pre>	<p>Sets the MED metric value.</p>
<p>Step 23 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 24 <code>route-map map-name [permit deny]</code> <code>[sequence-number]</code></p> <p>Example:</p> <pre>Router(config)# route-map BLOCK 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 BLOCK is created to block all incoming routes from autonomous system 45000.
<p>Step 25 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route map configuration mode and enters privileged EXEC mode.</p>
<p>Step 26 <code>show ip route [ip-address] [mask] [longer-prefixes]</code></p> <p>Example:</p> <pre>Router# show ip route</pre>	<p>(Optional) Displays route information from the routing tables.</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>.

Command or Action	Purpose
Step 27 <code>show ip bgp [network] [network-mask]</code> Example: <pre>Router# 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 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         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         0      32768 i
*> 172.17.1.0/24  0.0.0.0             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             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	<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> <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 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.

	Command or Action	Purpose
Step 7	<p>neighbor {<i>ip-address</i> <i>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>	<p>Exits address family configuration mode and enters router configuration mode.</p>
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>Router(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 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>} 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.

Command or Action	Purpose
<p>Step 12 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 13 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
<p>Step 14 <code>ip as-path access-list <i>access-list-number</i> {deny permit} <i>as-regular-expression</i></code></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.
<p>Step 15 <code>route-map <i>map-name</i> [permit deny][<i>sequence-number</i>]</code></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.
<p>Step 16 <code>match as-path <i>path-list-number</i></code></p> <p>Example:</p> <pre>Router(config-route-map)# match as-path 10</pre>	<p>Matches a BGP autonomous system path access list.</p> <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.
<p>Step 17 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route map configuration mode and enters privileged EXEC mode.</p>
<p>Step 18 <code>show ip bgp [<i>network</i>] [<i>network-mask</i>]</code></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      0  32768 i
```

Configuring BGP Policies

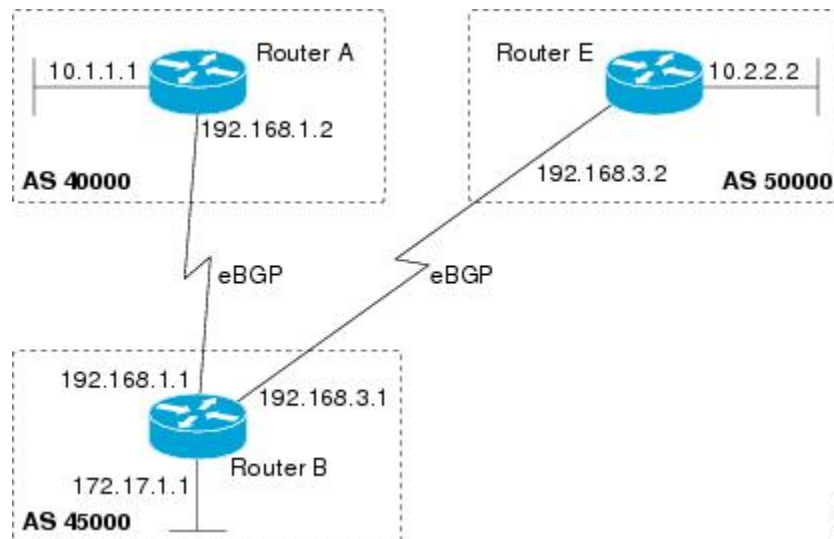
The tasks in this section help you configure BGP policies that filter the traffic in your BGP network. The following optional tasks demonstrate some of the various methods by which traffic can be filtered in your BGP network:

- [Filtering BGP Prefixes with Prefix Lists, page 169](#)
- [Filtering BGP Prefixes with AS-path Filters, page 173](#)
- [Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers, page 176](#)
- [Filtering Traffic Using Community Lists, page 180](#)
- [Filtering Traffic Using Extended Community Lists, page 185](#)
- [Filtering Traffic Using a BGP Route Map Policy List, page 189](#)
- [Filtering Traffic Using Continue Clauses in a BGP Route Map, page 193](#)

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 18 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	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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></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>
<p>Step 5 Repeat Step 5 for all BGP peers.</p>	<p>--</p>
<p>Step 6 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i></code></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>Step 7 <code>network <i>network-number</i> [mask <i>network-mask</i>]</code></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.
<p>Step 8 <code>aggregate-address <i>address mask</i> [as-set]</code></p> <p>Example:</p> <pre>Router(config-router-af)# aggregate- address 172.0.0.0 255.0.0.0</pre>	<p>Creates an aggregate entry in a BGP routing table.</p> <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. <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
<p>Step 9 <code>neighbor ip-address prefix-list list-name {in out}</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.2 prefix-list super172 out</pre>	<p>Distributes BGP neighbor information as specified in a prefix list.</p> <ul style="list-style-type: none"> In this example, a prefix list called super172 is set for outgoing routes to Router A.
<p>Step 10 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 11 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router) exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
<p>Step 12 <code>ip prefix-list list-name [seq seq-number] {deny network / length permit network / length} [ge ge-value] [le le-value] [eq eq-value]</code></p> <p>Example:</p> <pre>Router(config)# ip prefix-list super172 permit 172.0.0.0/8</pre>	<p>Defines a BGP-related prefix list and enters access list configuration mode.</p> <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.
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-access-list)# end</pre>	<p>Exits access list configuration mode and enters privileged EXEC mode.</p>
<p>Step 14 <code>show ip prefix-list [detail summary] [prefix-list-name [seq seq-number network / length] [longer first-match]]</code></p> <p>Example:</p> <pre>Router# show ip prefix-list detail super172</pre>	<p>Displays information about prefix lists.</p> <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
Step 15 <code>clear ip prefix-list {* ip-address peer-group-name} out</code>	Resets the hit count of the prefix list entries. <ul style="list-style-type: none"> In this example, the hit count for the prefix list called super172 will be reset.
Example: Router# <code>clear ip prefix-list super172 out</code>	

Examples

The following output from the `show ip prefix-list` command shows details of the prefix list named super172, including the hit count. The `clear ip prefix-list` command is entered to reset the hit count and the `show ip prefix-list` command is entered again to show the hit count reset to 0.

```
Router# show ip prefix-list detail super172
ip prefix-list super172:
  count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
  seq 5 permit 172.0.0.0/8 (hit count: 1, refcount: 1)
Router# clear ip prefix-list super172
Router# show ip prefix-list detail super172
ip prefix-list super172:
  count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
  seq 5 permit 172.0.0.0/8 (hit count: 0, refcount: 1)
```

Filtering BGP Prefixes with AS-path Filters

Perform this task to filter BGP prefixes using AS-path filters with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list is configured at Router B in the figure above. The first line of the access list denies all matches to the AS-path 50000 and the second line allows all other paths. The router uses the `neighbor filter-list` command to specify the AS-path access list as an outbound filter. After the filtering is enabled, traffic can be received from both Router A and Router 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 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.

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: 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	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Router(config-router)# neighbor 192.168.1.2 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.

Command or Action	Purpose
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>
Step 8 neighbor { <i>ip-address</i> <i>peer-group-name</i> } filter-list <i>access-list-number</i> { in out } Example: <pre>Router(config-router-af)# neighbor 192.168.1.2 filter-list 100 out</pre>	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 9 exit Example: <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode and returns to router configuration mode.
Step 10 exit Example: <pre>Router(config-router)# exit</pre>	Exits router configuration mode and returns to global configuration mode.

Command or Action	Purpose
<p>Step 11 <code>ip as-path access-list <i>access-list-number</i> {deny permit} <i>as-regular-expression</i></code></p> <p>Example:</p> <pre>Router(config)# ip as-path access-list 100 deny ^50000\$</pre> <p>Example:</p> <pre>Router(config)# ip as-path access-list 100 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 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. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
<p>Step 12 Repeat Step 11 for all entries required in the AS-path access list.</p>	--
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-access-list)# end</pre>	<p>Exits access list configuration mode and returns to privileged EXEC mode.</p>
<p>Step 14 <code>show ip bgp regexp <i>as-regular-expression</i></code></p> <p>Example:</p> <pre>Router# show ip bgp regexp ^50000\$</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 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:

```
Router# show ip bgp regexp ^50000$
BGP table version is 9, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * 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 [BGP Autonomous System Number Formats](#), page 128.

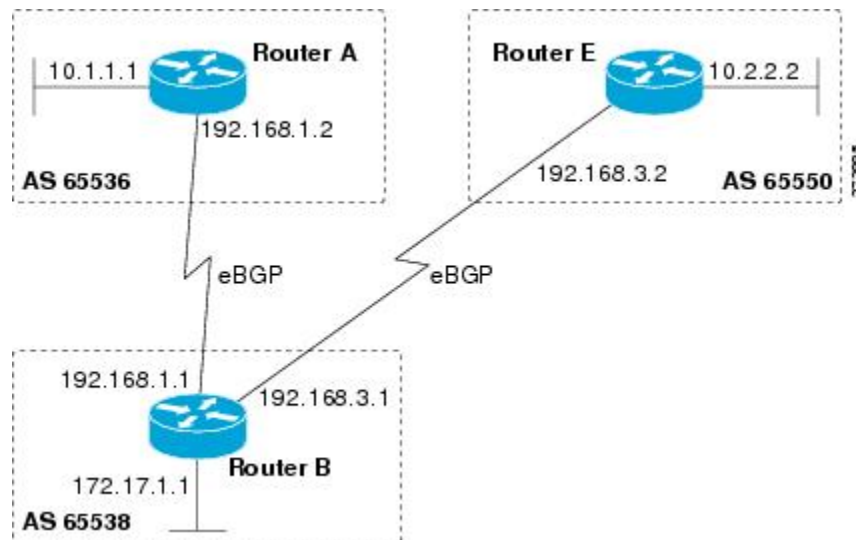
Perform this task to filter BGP prefixes with AS-path filters using 4-byte autonomous system numbers with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list

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 19 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: 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 65538	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: Router(config-router-af)# neighbor 192.168.1.2 remote-as 65536	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.

	Command or Action	Purpose
Step 5	Repeat Step 4 for all BGP peers.	--
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. <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 8	<p>neighbor {<i>ip-address</i> <i>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>

Command or Action	Purpose
<p>Step 11 <code>ip as-path access-list</code> <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> <p>and</p> <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>
<p>Step 12 Repeat Step 11 for all entries required in the AS-path access list.</p>	--
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-access-list)# end</pre>	<p>Exits access list configuration mode and returns to privileged EXEC mode.</p>
<p>Step 14 <code>show ip bgp regexp</code> <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.

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 BGP community lists and then reference them within a route map to control incoming routes. BGP communities provide a method of filtering inbound or outbound routes for large, complex networks. Instead of compiling long access or prefix lists of individual peers,

BGP allows grouping of peers with identical routing policies even though they reside in different autonomous systems or networks.

In this task, Router B in the figure above is configured with several route maps and community lists to control incoming routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number*
5. **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 15 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: 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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(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 BGP neighbor table of the local router.</p>
<p>Step 5 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>]</code></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>Step 6 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>route-map-name</i>{in out}</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 7 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 8 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 9 <code>route-map map-name [permit deny] [sequence-number]</code></p> <p>Example:</p> <pre>Router(config)# route-map 2000 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 2000 is defined.
<p>Step 10 <code>match community {standard-list-number expanded-list-number community-list-name [exact]}</code></p> <p>Example:</p> <pre>Router(config-route-map)# match community 1</pre>	<p>Matches a BGP community list.</p> <ul style="list-style-type: none"> In this example, the community attribute is matched to community list 1.
<p>Step 11 <code>set weight weight</code></p> <p>Example:</p> <pre>Router(config-route-map)# set weight 30</pre>	<p>Specifies the BGP weight for the routing table.</p> <ul style="list-style-type: none"> In this example, any route that matches community list 1 will have the BGP weight set to 30.
<p>Step 12 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 13 <code>route-map map-name [permit deny] [sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 14 <code>match community {standard-list-number expanded-list-number community-list-name [exact]}</code></p> <p>Example:</p> <pre>Router(config-route-map)# match community 2</pre>	<p>Matches a BGP community list.</p> <ul style="list-style-type: none"> In this example, the community attribute is matched to community list 2.

Command or Action	Purpose
<p>Step 15 <code>set community community-number</code></p> <p>Example:</p> <pre>Router(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 BGP community attribute set to 99.
<p>Step 16 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 17 <code>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}</code></p> <p>Example:</p> <pre>Router(config)# ip community-list 1 permit 100</pre> <p>Example:</p> <pre>and</pre> <p>Example:</p> <pre>Router(config)# ip community-list 2 permit internet</pre>	<p>Creates a community list for BGP and controls access to it.</p> <ul style="list-style-type: none"> In the first example, community list 1 permits routes with a community attribute of 100. Router C routes all have community attribute of 100 so their weight will be set to 30. In the second example, community list 2 effectively permits all routes by using the internet keyword. Any routes that did not match community list 1 are checked against community list 2. All routes are permitted but no changes are made to the route attributes. <p>Note Two examples are shown here because the task example requires both these statements to be configured.</p>
<p>Step 18 Repeat Step 15 to create all the required community lists.</p>	<p>--</p>
<p>Step 19 <code>exit</code></p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 20 <code>show ip community-list</code> [<i>standard-list-number</i> <i>expanded-list-number</i> <i>community-list-name</i>] [exact-match]</p> <p>Example:</p> <pre>Router# show ip community-list 1</pre>	<p>Displays configured BGP community list entries.</p>

Examples

The following sample output verifies that community list 1 has been created, with the output showing that community list 1 permits routes with a community attribute of 100:

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

The following sample output verifies that community list 2 has been created, with the output showing that community list 2 effectively permits all routes by using the **internet** keyword:

```
Router# show ip community-list 2
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. BGP communities provide a method of filtering inbound or outbound routes for large, complex networks. Instead of compiling long access or prefix lists of individual peers, BGP allows grouping of peers with identical routing policies even though they reside in different autonomous systems or networks.

In this task, Router B in the figure above is configured with an extended named community list to specify that the BGP peer at 192.168.1.2 is not sent advertisements about any path through or from autonomous system 50000. The IP extended community-list configuration mode is used and the ability to resequence entries is shown.



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.

>

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 Step 10 for all 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: 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 extcommunity-list { <i>expanded-list-number</i> expanded <i>list-name</i> <i>standard-list-number</i> standard <i>list-name</i> } Example: Router(config)# ip extcommunity-list expanded DENY50000	Enters IP extended community-list configuration mode to create or configure an extended community list. <ul style="list-style-type: none"> • In this example, the expanded community list DENY50000 is created.

Command or Action	Purpose
<p>Step 4 <code>[sequence-number] {deny[regular-expression] exit permit[regular-expression]}</code></p> <p>Example:</p> <pre>Router(config-extcomm-list)# 10 deny _50000_</pre> <p>Example:</p> <p>and</p> <p>Example:</p> <pre>Router(config-extcomm-list)# 20 deny ^50000 .*</pre>	<p>Configures an expanded community list entry.</p> <ul style="list-style-type: none"> In the first example, an expanded community list entry with the sequence number 10 is configured to deny advertisements about paths from autonomous system 50000. In the second example, an expanded community list entry with the sequence number 20 is configured to deny advertisements about paths through autonomous system 50000. <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>
<p>Step 5 Repeat Step 4 for all the required permit or deny entries in the extended community list.</p>	<p>--</p>
<p>Step 6 <code>resequence [starting-sequence][sequence-increment]</code></p> <p>Example:</p> <pre>Router(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>
<p>Step 7 <code>exit</code></p> <p>Example:</p> <pre>Router(config-extcomm-list)# exit</pre>	<p>Exits expanded community-list configuration mode and enters global configuration mode.</p>
<p>Step 8 <code>router bgp autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>

Command or Action	Purpose
<p>Step 9 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 192.168.3.2 remote-as 50000</pre>	<p>Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router.</p>
<p>Step 10 Repeat Step 10 for all the required BGP peers.</p>	<p>--</p>
<p>Step 11 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>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>
<p>Step 12 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. <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>
<p>Step 13 end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>
<p>Step 14 show ip extcommunity-list [<i>list-name</i>]</p> <p>Example:</p> <pre>Router# 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:

```
Router# 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.

BGP routing must be configured in your network and BGP neighbors must be established.



Note

- BGP route map policy lists do not support the configuration of IP version 6 (IPv6) match clauses in policy lists.
- Policy lists 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	<p>match metric <i>metric</i></p> <p>Example:</p> <pre>Router(config-policy-list)# match metric 10</pre>	Creates a match clause to permit routes with the specified metric.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-policy-list)# exit</pre>	Exits policy list configuration mode and enters global configuration mode.
Step 7	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map MAP-NAME-1 permit 10</pre>	Creates a route map and enters route map configuration mode.
Step 8	<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>Router(config-route-map)# match ip address 1</pre>	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	<p>match policy-list <i>policy-list-name</i></p> <p>Example:</p> <pre>Router(config-route-map)# match policy-list POLICY-LIST-NAME-1</pre>	<p>Creates a clause that will match the specified policy list.</p> <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	<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 10: 1</pre>	Creates a clause to set or remove the specified community.

Command or Action	Purpose
<p>Step 11 <code>set local-preference <i>preference-value</i></code></p> <p>Example:</p> <pre>Router(config-route-map)# set local-preference 140</pre>	Creates a clause to set the specified local preference value.
<p>Step 12 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	Exits route map configuration mode and enters privileged EXEC mode.
<p>Step 13 <code>show ip policy-list [<i>policy-list-name</i>]</code></p> <p>Example:</p> <pre>Router# show ip policy-list POLICY-LIST-NAME-1</pre>	Display information about configured policy lists and policy list entries.
<p>Step 14 <code>show route-map [<i>route-map-name</i>]</code></p> <p>Example:</p> <pre>Router# show route-map</pre>	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

Perform this task to filter 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* [**additive**] [*well-known-community*] | **none**}
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> 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: 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, and creates a BGP routing process.

Command or Action	Purpose
<p>Step 4 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 10.0.0.1 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 router.</p>
<p>Step 5 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>
<p>Step 6 neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>map-name</i>{in out}</p> <p>Example:</p> <pre>Router(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>
<p>Step 7 exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 8 exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
<p>Step 9 route-map <i>map-name</i> {permit deny} [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(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
<p>Step 10 <code>match ip address</code> {<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 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>
<p>Step 11 <code>set community</code> <i>community-number</i> [additive] [<i>well-known-community</i>] none}</p> <p>Example:</p> <pre>Router(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.
<p>Step 12 <code>continue</code> [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(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." <p>Note Continue clauses in outbound route maps are supported only in Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases.</p>
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route-map configuration mode and enters privileged EXEC mode.</p>
<p>Step 14 <code>show route-map</code> [<i>map-name</i>]</p> <p>Example:</p> <pre>Router# 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.

```
Router# show route-map
route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  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

- [Influencing Inbound Path Selection Examples, page 197](#)
- [Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers Example, page 197](#)
- [Influencing Outbound Path Selection Examples, page 199](#)
- [Filtering BGP Prefixes with Prefix Lists Examples, page 200](#)
- [Filtering Traffic Using Community Lists Examples, page 201](#)
- [Filtering Traffic Using AS-path Filters Example, page 202](#)
- [Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers Examples, page 202](#)
- [Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example, page 203](#)
- [Filtering Traffic Using a BGP Route Map Example, page 206](#)
- [Filtering Traffic Using Continue Clauses in a BGP Route Map Examples, page 206](#)

Influencing Inbound Path Selection Examples

The following example shows how you can use route maps to modify incoming data from a neighbor. Any route received from 10.222.1.1 that matches the filter parameters set in autonomous system access list 200 will have its weight set to 200 and its local preference set to 250, and it will be accepted.

```
router bgp 100
!
 neighbor 10.222.1.1 route-map FIX-WEIGHT in
 neighbor 10.222.1.1 remote-as 1
!
ip as-path access-list 200 permit ^690$
ip as-path access-list 200 permit ^1800
!
route-map FIX-WEIGHT permit 10
 match as-path 200
 set local-preference 250
 set weight 200
```

In the following example, the route map named finance marks all paths originating from autonomous system 690 with an MED metric attribute of 127. The second permit clause is required so that routes not matching autonomous system path list 1 will still be sent to neighbor 10.1.1.1.

```
router bgp 65000
 neighbor 10.1.1.1 route-map finance out
!
ip as-path access-list 1 permit ^690_
ip as-path access-list 2 permit .*
!
route-map finance permit 10
 match as-path 1
 set metric 127
!
route-map finance permit 20
 match as-path 2
```

Inbound route maps could perform prefix-based matching and set various parameters of the update. Inbound prefix matching is available in addition to autonomous system path and community list matching. The following example shows how the **set local-preference** route map configuration command sets the local preference of the inbound prefix 172.20.0.0/16 to 120:

```
!
router bgp 65100
 network 10.108.0.0
 neighbor 10.108.1.1 remote-as 65200
 neighbor 10.108.1.1 route-map set-local-pref in
!
route-map set-local-pref permit 10
 match ip address 2
 set local preference 120
!
route-map set-local-pref permit 20
!
access-list 2 permit 172.20.0.0 0.0.255.255
access-list 2 deny any
```

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

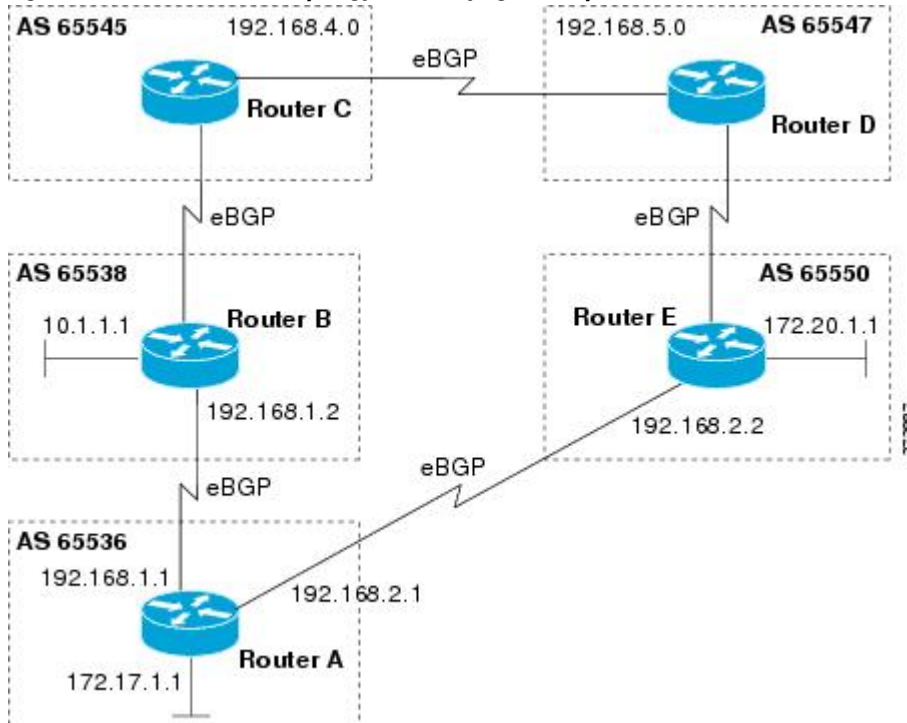
This example shows how to configure BGP to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SX11, and later releases, BGP support for 4-octet (4-byte) autonomous system

numbers was introduced. The 4-byte autonomous system numbers in this example are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure below. For more details about the introduction of 4-byte autonomous system numbers, see [BGP Autonomous System Number Formats](#), page 128.

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 20 Network Topology for Modifying the AS-path Attribute



The configuration for this example is performed at Router A in the figure above.

```
router bgp 65536
 address-family ipv4 unicast
  network 172.17.1.0 mask 255.255.255.0
  neighbor 192.168.1.2 remote-as 65538
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-map PREPEND out
 exit-address-family
exit
route-map PREPEND permit 10
set as-path prepend 65536 65536
```

Influencing Outbound Path Selection Examples

The following example creates an outbound route filter and configures Router-A (10.1.1.1) to advertise the filter to Router-B (172.16.1.2). An IP prefix list named FILTER is created to specify the 192.168.1.0/24 subnet for outbound route filtering. The ORF send capability is configured on Router-A so that Router-A can advertise the outbound route filter to Router-B.

Router-A Configuration (Sender)

```
ip prefix-list FILTER seq 10 permit 192.168.1.0/24
!
router bgp 65100
 address-family ipv4 unicast
  neighbor 172.16.1.2 remote-as 65200
  neighbor 172.16.1.2 ebgp-multihop
  neighbor 172.16.1.2 capability orf prefix-list send
  neighbor 172.16.1.2 prefix-list FILTER in
end
```

Router-B Configuration (Receiver)

The following example configures Router-B to advertise the ORF receive capability to Router-A. Router-B will install the outbound route filter, defined in the FILTER prefix list, after ORF capabilities have been exchanged. An inbound soft reset is initiated on Router-B at the end of this configuration to activate the outbound route filter.

```
router bgp 65200
 address-family ipv4 unicast
  neighbor 10.1.1.1 remote-as 65100
  neighbor 10.1.1.1 ebgp-multihop 255
  neighbor 10.1.1.1 capability orf prefix-list receive
end
clear ip bgp 10.1.1.1 in prefix-filter
```

The following example shows how the route map named set-as-path is applied to outbound updates to the neighbor 10.69.232.70. The route map will prepend the autonomous system path "65100 65100" to routes that pass access list 1. The second part of the route map is to permit the advertisement of other routes.

```
router bgp 65100
 network 172.16.0.0
 network 172.17.0.0
 neighbor 10.69.232.70 remote-as 65200
 neighbor 10.69.232.70 route-map set-as-path out
!
route-map set-as-path 10 permit
 match address 1
 set as-path prepend 65100 65100
!
route-map set-as-path 20 permit
 match address 2
```

```
!
access-list 1 permit 172.16.0.0 0.0.255.255
access-list 1 permit 172.17.0.0 0.0.255.255
!
access-list 2 permit 0.0.0.0 255.255.255.255
```

Filtering BGP Prefixes with Prefix Lists Examples

This section contains the following examples:

- [Filtering BGP Prefixes Using a Single Prefix List, page 200](#)
- [Filtering BGP Prefixes Using a Group of Prefixes, page 201](#)
- [Adding or Deleting Prefix List Entries, page 201](#)

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 GigabitEthernet interface 0/0/0:

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

Filtering BGP Prefixes Using a Group of Prefixes

The following example shows how to configure BGP to permit routes with a prefix length up to 24 in network 192/8:

```
ip prefix-list abc permit 192.0.0.0/8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in 192/8:

```
ip prefix-list abc deny 192.0.0.0/8 ge 25
```

The following example shows how to configure BGP to permit routes with a prefix length greater than 8 and less than 24 in all address space:

```
ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in all address space:

```
ip prefix-list abc deny 0.0.0.0/0 ge 25
```

The following example shows how to configure BGP to deny all routes in network 10/8, because any route in the Class A network 10.0.0.0/8 is denied if its mask is less than or equal to 32 bits:

```
ip prefix-list abc deny 10.0.0.0/8 le 32
```

The following example shows how to configure BGP to deny routes with a mask greater than 25 in 192.168.1.0/24:

```
ip prefix-list abc deny 192.168.1.0/24 ge 25
```

The following example shows how to configure BGP to permit all routes:

```
ip prefix-list abc permit 0.0.0.0/0 le 32
```

Adding or Deleting Prefix List Entries

You can add or delete individual entries in a prefix list if a prefix list has the following initial configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 192.168.0.0/15
```

The following example shows how to delete an entry from the prefix list so that 192.168.0.0 is not permitted, and add a new entry that permits 10.0.0.0/8:

```
no ip prefix-list abc permit 192.168.0.0/15
ip prefix-list abc permit 10.0.0.0/8
```

The new configuration is as follows:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 10.0.0.0/8
```

Filtering Traffic Using Community Lists Examples

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

The first example shows how the route map named `set-community` is applied to the outbound updates to the neighbor 172.16.232.50. The routes that pass access list 1 have the special community attribute value `no-export`. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```
router bgp 100
 neighbor 172.16.232.50 remote-as 200
 neighbor 172.16.232.50 send-community
 neighbor 172.16.232.50 route-map set-community out
!
route-map set-community permit 10
 match address 1
  set community no-export
!
route-map set-community permit 20
 match address 2
```

The second example shows how the route map named `set-community` is applied to the outbound updates to neighbor 172.16.232.90. All the routes that originate from autonomous system 70 have the community values 200 200 added to their already existing values. All other routes are advertised as normal.

```
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 .*
```

Filtering Traffic Using AS-path Filters Example

The following example shows BGP path filtering by neighbor. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.12.10. Similarly, only routes passing access list 3 will be accepted from 192.168.12.10.

```
router bgp 200
 neighbor 192.168.12.10 remote-as 100
 neighbor 192.168.12.10 filter-list 1 out
 neighbor 192.168.12.10 filter-list 2 in
 exit
ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```

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

Asplain Default Format in Cisco IOS 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)SXII, 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
```

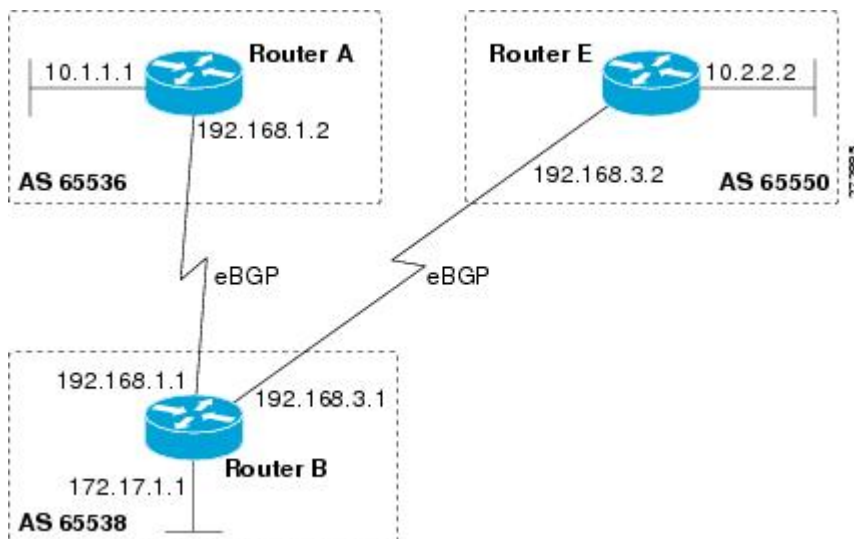
Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers Example

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXII, 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)SXII, 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 21 *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

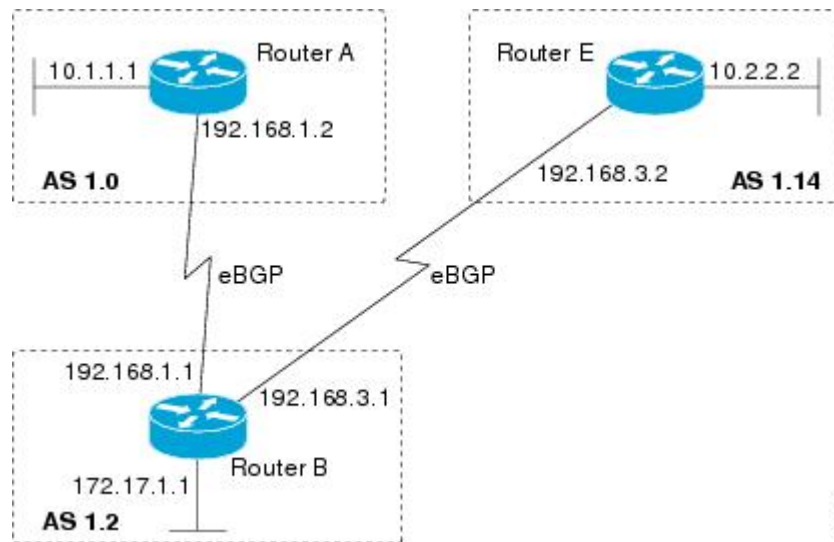
The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, extended BGP communities support 4-byte autonomous system numbers in the regular expressions in asdot format only. 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 22 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 192.1681.2 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 DENY114
 10 deny _1\.14_
 20 deny ^1\.14 .*
 resequence 50 100
 exit
router bgp 1.2
 network 172.17.1.0 mask 255.255.255.0
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.1.2 remote-as 1.0
  neighbor 192.168.3.2 activate
  neighbor 192.168.1.2 activate
 end
show ip extcommunity-list DENY114
```

Filtering Traffic Using a BGP Route Map Example

The following example shows how to use an address family to configure BGP so that any unicast and multicast routes from neighbor 10.1.1.1 are accepted if they match access list 1:

```
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
router bgp 65538
 neighbor 10.1.1.1 remote-as 65537
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
 end
```

Filtering Traffic Using Continue Clauses in a BGP Route Map Examples

The following example shows continue clause configuration in a route map sequence.



Note

Continue clauses in outbound route maps are supported only in Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful match occurs. If a match does not occur, the route map will "fall through" to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries. Only the first successful match ip address clause is supported.

If a successful match does not occur in route map entry 20, the route map will "fall through" to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed and the continue clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will "fall through" to route map entry 30 and execute the set clause. A sequence number is not specified for the continue clause so route map entry 40 will be evaluated.

There are two behaviors that can occur when the same **set** command is repeated in subsequent continue clause entries. For **set** commands that configure an additive or accumulative value (for example, **set community additive**, **set extended community additive**, and **set as-path prepend**), subsequent values are added by subsequent entries. The following example illustrates this behavior. After each set of match clauses, a **set as-path prepend** command is configured to add an autonomous system number to the as-path. After a match occurs, the route map stops evaluating match clauses and starts executing the set clauses, in the order in which they were configured. Depending on how many successful match clauses occur, the as-path is prepended by one, two, or three autonomous system numbers.

```
route-map ROUTE-MAP-NAME permit 10
 match ip address 1
 match metric 10
 set as-path prepend 10
 continue 30
 !
```

```

route-map ROUTE-MAP-NAME permit 20
  match ip address 2
  match metric 20
  set as-path prepend 10 10
  !
route-map ROUTE-MAP-NAME permit 30
  set as-path prepend 10 10 10
  continue
  !
route-map ROUTE-MAP-NAME permit 40
  match community 10:1
  set local-preference 104

```

In this example, the same **set** command is repeated in subsequent continue clause entries but the behavior is different from the first example. For **set** commands that configure an absolute value, the value from the last instance will overwrite the previous value(s). The following example illustrates this behavior. The set clause value in sequence 20 overwrites the set clause value from sequence 10. The next hop for prefixes from the 172.16/16 network is set to 10.2.2.2 and not 10.1.1.1.

```

ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
  match ip address prefix-list 1
  set ip next hop 10.1.1.1
  continue 20
  exit
route-map RED permit 20
  match ip address prefix-list 2
  set ip next hop 10.2.2.2
  end

```

**Note**

Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. The following example illustrates this case.

In the following example, when routes match an as-path of 10, 20, or 30, the routes are permitted and the continue clause jumps over the explicit deny clause to process the match ip address prefix list. If a match occurs here, the route metric is set to 100. Only routes that do not match an as-path of 10, 20, or 30 and do match a community number of 30 are denied. To deny other routes, you must configure an explicit deny statement.

```

route-map test permit 10
  match as-path 10 20 30
  continue 30
  exit
route-map test deny 20
  match community 30
  exit
route-map test permit 30
  match ip address prefix-list 1
  set metric 100
  exit

```

Where to Go Next

- To configure advanced BGP feature tasks, proceed to the "Configuring Advanced BGP Features" module.
- To configure BGP neighbor session options, proceed to the "Configuring BGP Neighbor Session Options" module.

- To configure internal BGP tasks, proceed to the "Configuring Internal BGP Features" module.

Additional References

The following sections provide references related to connecting to a service provider using external BGP.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
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
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	Title
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	A Border Gateway Protocol 4 (BGP-4)
RFC 4893	BGP Support for Four-Octet AS Number Space
RFC 5396	Textual Representation of Autonomous system (AS) Numbers
RFC 5398	Autonomous System (AS) Number Reservation for Documentation Use

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/cisco/web/support/index.html</p>

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16 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 Prefix-Based Outbound Route Filtering	12.0(22)S 12.2(4)T 12.2(14)S 15.0(1)S	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	12.0(24)S 12.2(18)S 12.2(18)SXD 12.2(27)SBC 12.3(2)T 15.0(1)S Cisco IOS XE 3.1.0SG	The BGP Route-Map Continue feature introduces the continue clause to BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map.
BGP Route-Map Continue Support for an Outbound Policy	12.0(31)S 12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(4)T 15.0(1)S Cisco IOS XE 3.1.0SG	The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps.

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	<p>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.</p>

Feature Name	Releases	Feature Configuration Information
BGP Support for 4-Byte ASN	12.0(32)S12 12.0(32)SY8 12.0(33)S3 12.2(33)SRE 12.2(33)XNE 12.2(33)SXII 12.4(24)T 15.0(1)S Cisco IOS XE 3.1.0SG	<p>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.</p> <p>In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, and 12.2(33)SXII, the Cisco implementation of 4-byte autonomous system numbers uses <code>asplain</code> as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the <code>asplain</code> format and the <code>asdot</code> format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to <code>asdot</code> format, use the <code>bgp asnotation dot</code> command.</p> <p>In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses <code>asdot</code> as the only configuration format, regular expression match, and output display, with no <code>asplain</code> support.</p> <p>The following commands were introduced or modified by this feature: <code>bgp asnotation dot</code>, <code>bgp confederation identifier</code>, <code>bgp confederation peers</code>, all <code>clear ip bgp</code> commands that configure an autonomous system number, <code>ip as-path access-list</code>, <code>ip extcommunity-list</code>, <code>match source-protocol</code>, <code>neighbor local-</code></p>

Feature Name	Releases	Feature Configuration Information
		as, neighbor remote-as, neighbor soo, redistribute (IP), router bgp, route-target, set as-path, set extcommunity, set origin, soo, all show ip bgp commands that display an autonomous system number, and show ip extcommunity-list .
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	

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: 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)

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



Configuring BGP Neighbor Session Options

This module describes configuration tasks to configure various options involving Border Gateway Protocol (BGP) neighbor peer sessions. BGP is an interdomain routing protocol designed to provide loop-free routing between organizations. This module contains tasks that use BGP neighbor session commands to configure:

- Fast session deactivation
- Bidirectional Forwarding Detection (BFD) for BGP IPv6 neighbors
- A router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down
- Options to help an autonomous system migration
- TTL Security Check, a lightweight security mechanism to protect External BGP (eBGP) peering sessions from CPU-utilization-based attacks
- [Finding Feature Information, page 215](#)
- [Prerequisites for Configuring BGP Neighbor Session Options, page 215](#)
- [Restrictions for Configuring BGP Neighbor Session Options, page 216](#)
- [Information About Configuring BGP Neighbor Session Options, page 216](#)
- [How to Configure BGP Neighbor Session Options, page 222](#)
- [Configuration Examples for BGP Neighbor Session Options, page 255](#)
- [Where to Go Next, page 261](#)
- [Additional References, page 261](#)
- [Feature Information for Configuring BGP Neighbor Session Options, page 263](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Configuring BGP Neighbor Session Options

Before configuring advanced BGP features you should be familiar with the "Cisco BGP Overview" module and the "Configuring a Basic BGP Network" module.

Restrictions for Configuring BGP Neighbor Session Options

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

Information About Configuring BGP Neighbor Session Options

- [BGP Neighbor Sessions, page 216](#)
- [BGP Support for Fast Peering Session Deactivation, page 216](#)
- [BFD Support of BGP IPv6 Neighbors, page 217](#)
- [BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached, page 217](#)
- [BGP Network Autonomous System Migration, page 218](#)
- [TTL Security Check for BGP Neighbor Sessions, page 219](#)
- [BGP Support for TCP Path MTU Discovery per Session, page 221](#)
- [BGP Dynamic Neighbors, page 222](#)

BGP Neighbor Sessions

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. A BGP-speaking router does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking routers.

A BGP neighbor device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a peer instead of neighbor because a neighbor may imply the idea that the BGP devices are directly connected with no other router in between. Configuring BGP neighbor or peer sessions uses BGP neighbor session commands so this module will prefer the use of the term "neighbor" over "peer."

BGP Support for Fast Peering Session Deactivation

- [BGP Hold Timer, page 216](#)
- [BGP Fast Peering Session Deactivation, page 216](#)
- [Selective Address Tracking for BGP Fast Session Deactivation, page 217](#)

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco IOS software. This timer value is set as the default to protect the BGP routing process from instability that can be caused by peering sessions with other routing protocols. BGP routers typically carry large routing tables, so frequent session resets are not desirable.

BGP Fast Peering Session Deactivation

BGP fast peering session deactivation improves BGP convergence and response time to adjacency changes with BGP neighbors. This feature is event driven and configured on a per-neighbor basis. When this feature is enabled, BGP will monitor the peering session with the specified neighbor. Adjacency changes are detected and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

Selective Address Tracking for BGP Fast Session Deactivation

In Cisco IOS 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.

BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached

- [Prefix Limits and BGP Peering Sessions, page 217](#)
- [BGP Neighbor Session Restart with the Maximum Prefix Limit, page 218](#)

Prefix Limits and BGP Peering Sessions

There is a configurable limit on the maximum number of prefixes that a router that is running BGP can receive from a peer router. This limit is configured with the **neighbor maximum-prefix** command. When the router receives too many prefixes from a peer router and the maximum-prefix limit is exceeded, the peering session is disabled or brought down. The session stays down until the network operator manually brings the session back up by entering the **clear ip bgp** command. Entering the **clear ip bgp** command clears stored prefixes.

BGP Neighbor Session Restart with the Maximum Prefix Limit

In Cisco IOS Release 12.0(22)S, 12.2(15)T, 12.2(18)S, and later releases, the **restart** keyword was added to enhance the capabilities of the **neighbor maximum-prefix** command. This enhancement allows the network operator to configure a router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down. There is configurable time interval at which peering can be reestablished automatically. The configurable timer argument for the **restart** keyword is specified in minutes. The time range is from 1 to 65,535 minutes.

BGP Network Autonomous System Migration

- [Autonomous System Migration for BGP Networks, page 218](#)
- [Dual Autonomous System Support for BGP Network Autonomous System Migration, page 218](#)
- [BGP Network Migration to 4-Byte Autonomous System Numbers, page 219](#)

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 a group peers, the individual peers cannot be customized.

Ingress Filtering During BGP Autonomous System Migration

Autonomous system path customization increases the possibility that routing loops can be created if such customization is misconfigured. The larger the number of customer peerings, the greater the risk. You can minimize this possibility by applying policies on the ingress interfaces to block the autonomous system number that is in transition or routes that have no **local-as** configuration.

**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 will start 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.

TTL Security Check for BGP Neighbor Sessions

- [BGP Support for the TTL Security Check](#), page 219
- [TTL Security Check for BGP Neighbor Sessions](#), page 220
- [TTL Security Check Support for Multihop BGP Neighbor Sessions](#), page 220
- [Benefits of the BGP Support for TTL Security Check](#), page 220

BGP Support for the TTL Security Check

When implemented for BGP, the TTL Security Check feature introduces a lightweight security mechanism to protect eBGP neighbor sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses.

The TTL Security Check feature protects the eBGP neighbor session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP neighbor session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is unnecessary.

Although it is possible to forge the TTL field in an IP packet header, accurately forging the TTL count to match the TTL count from a trusted peer is impossible unless the network to which the trusted peer belongs has been compromised.

The TTL Security Check feature supports both directly connected neighbor sessions and multihop eBGP neighbor sessions. The BGP neighbor session is not affected by incoming packets that contain invalid TTL values. The BGP neighbor session will remain open, and the router will silently discard the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

TTL Security Check for BGP Neighbor Sessions

The BGP Support for TTL Security Check feature is configured with the **neighbor ttl-security** command in router configuration mode or address family configuration mode. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. The *hop-count* argument is used to configure the maximum number of hops that separate the two peers. The TTL value is determined by the router from the configured hop count. The value for this argument is a number from 1 to 254.

TTL Security Check Support for Multihop BGP Neighbor Sessions

The BGP Support for TTL Security Check feature supports both directly connected neighbor sessions and multihop neighbor sessions. When this feature is configured for a multihop neighbor session, the **neighbor ebgp-multihop** router configuration command cannot be configured and is not needed to establish the neighbor session. These commands are mutually exclusive, and only one command is required to establish a multihop neighbor session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.

To configure this feature for an existing multihop session, you must first disable the existing neighbor session with the **no neighbor ebgp-multihop** command. The multihop neighbor session will be restored when you enable this feature with the **neighbor ttl-security** command.

This feature should be configured on each participating router. To maximize the effectiveness of this feature, the *hop-count* argument should be strictly configured to match the number of hops between the local and external network. However, you should also consider path variation when configuring this feature for a multihop neighbor session.

Benefits of the BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect eBGP neighbor sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP autonomous system.

BGP Support for TCP Path MTU Discovery per Session

- [Path MTU Discovery, page 221](#)
- [BGP Neighbor Session TCP PMTUD, page 221](#)

Path MTU Discovery

The IP protocol family was designed to use a wide variety of transmission links. The maximum IP packet length is 65000 bytes. Most transmission links enforce a smaller maximum packet length limit, called the maximum transmission unit (MTU), which varies with the type of the transmission link. The design of IP accommodates link packet length limits by allowing intermediate routers to fragment IP packets as necessary for their outgoing links. The final destination of an IP packet is responsible for reassembling its fragments as necessary.

All TCP sessions are bounded by a limit on the number of bytes that can be transported in a single packet, and this limit is known as the maximum segment size (MSS). TCP breaks up packets into chunks in a transmit queue before passing packets down to the IP layer. A smaller MSS may not be fragmented at an IP device along the path to the destination device, but smaller packets increase the amount of bandwidth needed to transport the packets. The maximum TCP packet length is determined by both the MTU of the outbound interface on the source device and the MSS announced by the destination device during the TCP setup process.

Path MTU discovery (PMTUD) was developed as a solution to the problem of finding the optimal TCP packet length. PMTUD is an optimization (detailed in RFC 1191) wherein a TCP connection attempts to send the longest packets that will not be fragmented along the path from source to destination. It does this by using a flag, don't fragment (DF), in the IP packet. This flag is supposed to alter the behavior of an intermediate router that cannot send the packet across a link because it is too long. Normally the flag is off, and the router should fragment the packet and send the fragments. If a router tries to forward an IP datagram, with the DF bit set, to a link that has a lower MTU than the size of the packet, the router will drop the packet and return an ICMP Destination Unreachable message to the source of this IP datagram, with the code indicating "fragmentation needed and DF set." When the source device receives the ICMP message, it will lower the send MSS, and when TCP retransmits the segment, it will use the smaller segment size.

BGP Neighbor Session TCP PMTUD

TCP path MTU discovery is enabled by default for all BGP neighbor sessions, but there are situations when you may want to disable TCP path MTU discovery for one or all BGP neighbor sessions. Although PMTUD works well for larger transmission links (for example, Packet over Sonet links), a badly configured TCP implementation or a firewall may slow or stop the TCP connections from forwarding any packets. In this type of situation, you may need to disable TCP path MTU discovery. In Cisco IOS Release 12.2(33)SRA, 12.2(31)SB, 12.2(33)SXH, 12.4(20)T, and later releases, configuration options were introduced to permit TCP path MTU discovery to be disabled, or subsequently reenabled, either for a single BGP neighbor session or for all BGP sessions. To disable the TCP path MTU discovery globally for all BGP neighbors, use the **no bgp transport path-mtu-discovery** command in router configuration mode. To disable the TCP path MTU discovery for a single neighbor, use the **no neighbor transport path-mtu-discovery** command in router or address family configuration modes. For more details, see the [Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 239](#) or the [Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 241](#).

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 Neighbor Session Options

- [Configuring Fast Session Deactivation](#), page 222
- [Configuring BFD for BGP IPv6 Neighbors](#), page 226
- [Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded](#), page 229
- [Configuring Dual-AS Peering for Network Migration](#), page 233
- [Configuring the TTL Security Check for BGP Neighbor Sessions](#), page 235
- [Configuring BGP Support for TCP Path MTU Discovery per Session](#), page 239
- [Implementing BGP Dynamic Neighbors Using Subnet Ranges](#), page 248

Configuring Fast Session Deactivation

The tasks in this section show how to configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about route dampening, see the "Configuring Internal BGP Features" module.

- [Configuring Fast Session Deactivation for a BGP Neighbor](#), page 222
- [Configuring Selective Address Tracking for Fast Session Deactivation](#), page 224

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
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
<p>Step 4 address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</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 an IPv4 unicast address family session.
<p>Step 5 neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.1 remote-as 50000</pre>	<p>Establishes a peering session with a BGP neighbor.</p>

Command or Action	Purpose
Step 6 <code>neighbor ip-address fall-over</code> Example: <pre>Router(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 <code>end</code> Example: <pre>Router(config-router-af)# end</pre>	Exits 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	<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> <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>neighbor <i>ip-address</i> fall-over [route-map <i>map-name</i>]</p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR</pre>	<p>Applies a route map when a route to the BGP changes.</p> <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.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 7 <code>ip prefix-list list-name [seq seq-value]{deny network / length permit network / length}[ge ge-value] [le le-value]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 8 <code>route-map map-name [permit deny][sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 9 <code>match ip address prefix-list prefix-list-name [prefix-list-name...]</code></p> <p>Example:</p> <pre>Router(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>
<p>Step 10 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route-map configuration mode and enters privileged EXEC mode.</p>

- [What to Do Next, page 226](#)

What to Do Next

The BGP Support for Next-Hop Address Tracking feature improves the response time of BGP to next-hop changes for routes installed in the RIB, which can also improve overall BGP convergence. For information about BGP next-hop address tracking, see the "Configuring Advanced BGP Features" module.

Configuring BFD for BGP IPv6 Neighbors

In Cisco IOS 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** | **vpn6**]
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	<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>ipv6 unicast-routing</p> <p>Example:</p> <pre>Router(config)# ipv6 unicast-routing</pre>	<p>Enables the forwarding of IPv6 unicast datagrams.</p>
Step 4	<p>ipv6 cef</p> <p>Example:</p> <pre>Router(config)# ipv6 cef</pre>	<p>Enables Cisco Express Forwarding for IPv6.</p>

Command or Action	Purpose
<p>Step 5 <code>interface type number</code></p> <p>Example:</p> <pre>Router(config)# interface fastethernet 0/1</pre>	Configures an interface type and number.
<p>Step 6 <code>ipv6 address ipv6-address / prefix-length</code></p> <p>Example:</p> <pre>Router(config-if)# ipv6 address 2001:DB8:1:1::1/64</pre>	Configures an IPv6 address and enables IPv6 processing on an interface.
<p>Step 7 <code>bfd interval milliseconds min_rx milliseconds multiplier multiplier-value</code></p> <p>Example:</p> <pre>Router(config-if)# bfd interval 500 min_rx 500 multiplier 3</pre>	Sets the baseline BFD session parameters on an interface.
<p>Step 8 <code>no shutdown</code></p> <p>Example:</p> <pre>Router(config-if)# no shutdown</pre>	Restarts an interface.
<p>Step 9 <code>exit</code></p> <p>Example:</p> <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
<p>Step 10 <code>router bgp autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	Enters router configuration mode for the specified routing process.
<p>Step 11 <code>no bgp default ipv4-unicast</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the default IPv4 unicast address family for establishing peering sessions.</p> <ul style="list-style-type: none"> We recommend configuring this command in the global scope.

Command or Action	Purpose
<p>Step 12 <code>address-family ipv6 [vrf vrf-name] [unicast multicast vpnv6]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv6</pre>	<p>Enters address family configuration mode and enables IPv6 addressing.</p>
<p>Step 13 <code>neighbor ipv6-address remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 2001:DB8:2:1::4 remote-as 45000</pre>	<p>Adds the IP address of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table of the local router.</p>
<p>Step 14 <code>neighbor ipv6-address fall-over bfd</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 2001:DB8:2:1::4 fall- over bfd</pre>	<p>Enables BGP to monitor the peering session of an IPv6 neighbor using BFD.</p>
<p>Step 15 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits configuration mode and enters privileged EXEC mode.</p>

Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded

Perform this task to configure the time interval at which a BGP neighbor session is reestablished by a router when the number of prefixes that have been received from a BGP peer has exceeded the maximum prefix limit.

The network operator can configure a router that is running BGP to automatically reestablish a neighbor session that has been brought down because the configured maximum-prefix limit has been exceeded. No intervention from the network operator is required when this feature is enabled.

**Note**

This task attempts to reestablish a disabled BGP neighbor session at the configured time interval that is specified by the network operator. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. The network operator will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the **warning-only** keyword of the **neighbor maximum-prefix** command can be configured to disable the restart capability, while the network operator corrects the underlying problem.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *minutes*] [**warning-only**]
5. **end**
6. **show ip bgp neighbors** *ip-address*

DETAILED STEPS

	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.

Command or Action	Purpose
<p>Step 4 <code>neighbor {ip-address peer-group-name} maximum-prefix maximum [threshold] [restart minutes] [warning-only]</code></p> <p>Example:</p> <pre>Router(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 router to disable the restart capability to allow you to fix 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>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits configuration mode and enters privileged EXEC mode.</p>
<p>Step 6 <code>show ip bgp neighbors ip-address</code></p> <p>Example:</p> <pre>Router# 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 example output from the **show ip bgp neighbors** command verifies that a router has been configured to automatically reestablish disabled neighbor sessions. The output shows that the maximum prefix limit for neighbor 10.4.9.5 is set to 1000 prefixes, the restart threshold is set to 90 percent, and the restart interval is set at 60 minutes.

```
Router# show ip bgp neighbors 10.4.9.5

BGP neighbor is 10.4.9.5, remote AS 101, internal link
BGP version 4, remote router ID 10.4.9.5
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

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 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 by

```

- [Troubleshooting Tips, page 232](#)

Troubleshooting Tips

Use the **clear ip bgp** command to resets a BGP connection using BGP soft reconfiguration. This command can be used to clear stored prefixes to prevent a router that is running BGP from exceeding the maximum-prefix limit. For more details about using BGP soft reconfiguration, see the "Monitoring and Maintaining Basic BGP" task in the "Configuring a Basic BGP Network" module.

Display of the following error messages can indicate an underlying problem that is causing the neighbor session to become disabled. The network operator should check the values that are configured for the maximum-prefix limit and the configuration of any peers that are sending an excessive number of prefixes. The following sample error messages are similar to the error messages that may be displayed:

```

00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Up
00:01:14:%BGP-4-MAXPFX:No. of unicast prefix received from 10.10.10.2 reaches 5, max 6
00:01:14:%BGP-3-MAXPFXEXCEED:No.of unicast prefix received from 10.10.10.2:7 exceed limit6
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Down - BGP Notification sent
00:01:14:%BGP-3-NOTIFICATION:sent to neighbor 10.10.10.2 3/1 (update malformed) 0 byte

```

The **bgp dampening** command can be used to configure the dampening of a flapping route or interface when a peer is sending too many prefixes and causing network instability. Use this command only when

troubleshooting or tuning a router that is sending an excessive number of prefixes. For more details about BGP route dampening, see the "Configuring Advanced BGP Features" module.

Configuring Dual-AS Peering for Network Migration

Perform this task to configure a BGP peer router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. When the BGP peer is configured with dual autonomous system numbers then the network operator can merge a secondary autonomous system into a primary autonomous system and update the customer configuration during a future service window without disrupting existing peering arrangements.

The **show ip bgp** and **show ip bgp neighbors** commands can be used to verify autonomous system number for entries in the routing table and the status of this feature.



Note

- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for only true eBGP peering sessions. This feature cannot be configured for two peers in different subautonomous systems of a confederation.
- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for individual peering sessions and configurations applied through peer groups and peer templates. If this command is applied to a peer group, the peers cannot be individually customized.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **local-as** [*autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]]
6. **neighbor** *ip-address* **remove-private-as**
7. **end**
8. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter-prefixes** *mask-length*]
9. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes**| **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*]

DETAILED STEPS

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
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	<p>Enters router configuration mode, and creates a BGP routing process.</p>
<p>Step 4 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 remote-as 45000</pre>	<p>Establishes a peering session with a BGP neighbor.</p>
<p>Step 5 <code>neighbor <i>ip-address</i> local-as [<i>autonomous-system-number</i> no-prepend [replace-as [dual-as]]]</code></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 (local-as). • The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the local-as number.
<p>Step 6 <code>neighbor <i>ip-address</i> remove-private-as</code></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.

Command or Action	Purpose
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	Exits configuration mode and enters privileged EXEC mode.
<p>Step 8 <code>show ip bgp [network] [network-mask] [longer-prefixes] [prefix-list prefix-list-name route-map route-map-name] [shorter-prefixes mask-length]</code></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 local-as number is configured.
<p>Step 9 <code>show ip bgp neighbors [neighbor-address] [received-routes routes advertised-routes paths regexp dampened-routes received prefix-filter]</code></p> <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.

Configuring the TTL Security Check for BGP Neighbor Sessions

Perform this task to allow BGP to establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the BGP neighbor session.

- To maximize the effectiveness of the BGP Support for TTL Security Check feature, we recommend that you configure it on each participating router. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router.



Note

- The **neighbor ebgp-multihop** command is not needed when the BGP Support for TTL Security Check feature is configured for a multihop neighbor session and should be disabled before configuring this feature.
- The effectiveness of the BGP Support for TTL Security Check feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected neighbor sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of the local and remote network. This restriction also includes peers that are on the network segment between the local and remote network.

SUMMARY STEPS

1. **enable**
2. **trace** *[protocol] destination*
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *ip-address* **ttl-security hops** *hop-count*
6. **end**
7. **show running-config**
8. **show ip bgp neighbors** *[ip-address]*

DETAILED STEPS

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 trace <i>[protocol] destination</i> Example: <pre>Router# trace ip 10.1.1.1</pre>	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.
Step 3 configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 4 router bgp <i>autonomous-system-number</i> Example: <pre>Router(config)# router bgp 65000</pre>	Enters router configuration mode, and creates a BGP routing process.

Command or Action	Purpose
<p>Step 5 <code>neighbor ip-address ttl-security hops hop-count</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits configuration mode and enters privileged EXEC mode.</p>
<p>Step 7 <code>show running-config</code></p> <p>Example:</p> <pre>Router# 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>
<p>Step 8 <code>show ip bgp neighbors [ip-address]</code></p> <p>Example:</p> <pre>Router# 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 enabled. The <i>number</i> value represents the hop count. It is a number from 1 to 254. <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, page 239](#)
- [Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 241](#)
- [Enabling TCP Path MTU Discovery Globally for All BGP Sessions, page 244](#)
- [Enabling TCP Path MTU Discovery for a Single BGP Neighbor, page 246](#)

Disabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to disable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but we recommend that you enter the **show ip bgp neighbors** command to ensure that TCP path MTU discovery is enabled.

This task assumes that you have previously configured BGP neighbors with active TCP connections.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address*]
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **no bgp transport path-mtu-discovery**
6. **end**
7. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

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
<p>Step 2 <code>show ip bgp neighbors [ip-address]</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <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>
<p>Step 3 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 4 <code>router bgp autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
<p>Step 5 <code>no bgp transport path-mtu-discovery</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp transport path-mtu-discovery</pre>	<p>Disables TCP path MTU discovery for all BGP sessions.</p>
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and returns to privileged EXEC mode.</p>
<p>Step 7 <code>show ip bgp neighbors [ip-address]</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors</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 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 or address family configuration mode.

This task assumes that you know that TCP path MTU discovery is enabled by default for all your BGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** { **ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **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: <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 address-family { ipv4 [mdt multicast unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] vpn4 [unicast]} Example: <pre>Router(config-router)# address-family ipv4 unicast</pre>	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.

Command or Action	Purpose
<p>Step 5 <code>neighbor {ip-address peer-group-name} remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 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>
<p>Step 6 <code>neighbor {ip-address peer-group-name} activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 activate</pre>	<p>Activates the neighbor under the IPv4 address family.</p>
<p>Step 7 <code>no neighbor {ip-address peer-group-name} transport{connection-mode path-mtu-discovery}</code></p> <p>Example:</p> <pre>Router(config-router-af)# no neighbor 172.16.1.1 transport path-mtu-discovery</pre>	<p>Disables TCP path MTU discovery for a single BGP neighbor.</p> <ul style="list-style-type: none"> In this example, TCP path MTU discovery is disabled for the neighbor at 172.16.1.1.
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
<p>Step 9 <code>show ip bgp neighbors</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors</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 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>

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.

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
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></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>
<p>Step 4 <code>bgp transport path-mtu-discovery</code></p> <p>Example:</p> <pre>Router(config-router)# bgp transport path-mtu-discovery</pre>	<p>Enables TCP path MTU discovery for all BGP sessions.</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and returns to privileged EXEC mode.</p>
<p>Step 6 <code>show ip bgp neighbors</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors</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 show that all 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>

Examples

The following sample output from the `show ip bgp neighbors` command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
```

```

BGP neighbor is 172.16.1.2, remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
    
```

Enabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an eBGP neighbor and then enable TCP path MTU discovery for the BGP neighbor session. The **neighbor transport** command can be used in router configuration or address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** { **ipv4** [**mdt** | **multicast** | **unicast** [**vrf vrf-name**] | **vrf vrf-name**] | **vpnv4** [**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: 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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 <code>address-family {ipv4 [<i>mdt</i> <i>multicast</i> <i>unicast</i> [<i>vrf vrf-name</i>] <i>vrf vrf-name</i>] <i>vpn</i>v4 [<i>unicast</i>]}</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</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 an IPv4 unicast address family session.
<p>Step 5 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.2 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>
<p>Step 6 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.2 activate</pre>	<p>Activates the neighbor under the IPv4 address family.</p>
<p>Step 7 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} transport{<i>connection-mode</i> <i>path-mtu-discovery</i>}</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.2 transport path-mtu-discovery</pre>	<p>Enables TCP path MTU discovery for a single BGP neighbor.</p>
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Command or Action	Purpose
Step 9 <code>show ip bgp neighbors [ip-address]</code> Example: Router# <code>show ip bgp neighbors 192.168.2.2</code>	(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
```

Implementing BGP Dynamic Neighbors Using Subnet Ranges

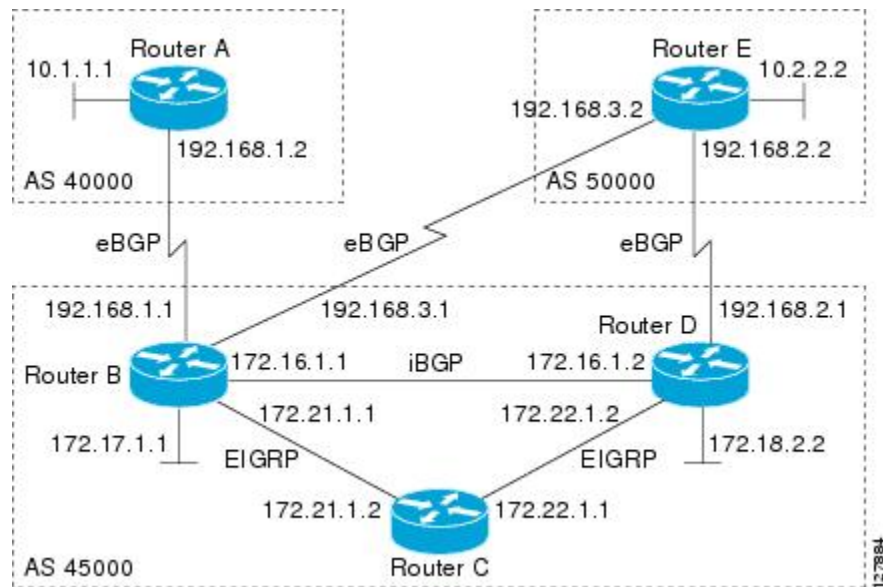
In Cisco IOS Release 12.2(33)SXH, support for BGP dynamic neighbors was introduced. Perform this task to implement the dynamic creation of BGP neighbors using subnet ranges.

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 23 *BGP Dynamic Neighbor Topology*



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: RouterB> 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: RouterB# 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>RouterB(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>RouterB(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>RouterB(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>RouterB(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>RouterB(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>ttl</i>]</p> <p>Example:</p> <pre>RouterB(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
<p>Step 9 <code>neighbor peer-group-name remote-as autonomous-system-number [alternate-as autonomous-system-number...]</code></p> <p>Example:</p> <pre>RouterB(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>
<p>Step 10 <code>address-family ipv4 [mdt multicast unicast [vrf vrf-name]]</code></p> <p>Example:</p> <pre>RouterB(config-router)# address-family ipv4 unicast</pre>	<p>Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.</p>
<p>Step 11 <code>neighbor {ip-address peer-group-name} activate</code></p> <p>Example:</p> <pre>RouterB(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>
<p>Step 12 <code>end</code></p> <p>Example:</p> <pre>RouterB(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
<p>Step 13 Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.</p>	<p>--</p>
<p>Step 14 <code>enable</code></p> <p>Example:</p> <pre>RouterE> 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.

Command or Action	Purpose
<p>Step 15 configure terminal</p> <p>Example:</p> <pre>RouterE# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 16 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>RouterE(config)# router bgp 50000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>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>]</p> <p>Example:</p> <pre>RouterE(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 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.
<p>Step 18 Return to the first router.</p>	<p>--</p>
<p>Step 19 show ip bgp summary</p> <p>Example:</p> <pre>RouterB# show ip bgp summary</pre>	<p>(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors.</p> <ul style="list-style-type: none"> In this step, the configuration has returned to Router B.
<p>Step 20 show ip bgp peer-group [<i>peer-group-name</i>] [summary]</p> <p>Example:</p> <pre>RouterB# show ip bgp peer-group group192</pre>	<p>(Optional) Displays information about BGP peer groups.</p>

Command or Action	Purpose
<p>Step 21 <code>show ip bgp neighbors [ip-address]</code></p> <p>Example:</p> <pre>RouterB# show ip bgp neighbors 192.168.3.2</pre>	<p>(Optional) Displays information about BGP and TCP connections to neighbors.</p> <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 peer group 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 peer group 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 Neighbor Session Options

- [Example Configuring Fast Session Deactivation for a BGP Neighbor, page 255](#)
- [Example Configuring Selective Address Tracking for Fast Session Deactivation, page 256](#)
- [Example Configuring BFD for a BGP IPv6 Neighbor, page 256](#)
- [Example Restart Session After Maximum Number of Prefixes From Neighbor Reached, page 256](#)
- [Configuration Examples for Dual AS Peering for Network Migration, page 256](#)
- [Example Configuring the TTL-Security Check, page 258](#)
- [Examples Configuring BGP Support for TCP Path MTU Discovery per Session, page 258](#)
- [Example Implementing BGP Dynamic Neighbors Using Subnet Ranges, page 259](#)

Example Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on Router A and Router B to monitor and use fast peering session deactivation for the neighbor session between the two routers. Although fast peering session deactivation is not required at both routers in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Router A

```

router bgp 40000
neighbor 192.168.1.1 remote-as 45000
neighbor 192.168.1.1 fall-over
end

```

Router 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 Restart Session After Maximum Number of Prefixes From Neighbor Reached

The following example sets the maximum number of prefixes allowed from the neighbor at 192.168.6.6 to 2000 and configures the router to reestablish a peering session after 30 minutes if one has been disabled:

```
router bgp 101
 network 172.16.0.0
 neighbor 192.168.6.6 maximum-prefix 2000 restart 30
```

Configuration Examples for Dual AS Peering for Network Migration

- [Example: Dual AS Configuration, page 256](#)
- [Example: Dual AS Confederation Configuration, page 257](#)
- [Example: Replace an AS with Another AS in Routing Updates, page 257](#)

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
```

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, page 258](#)
- [Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 258](#)
- [Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions, page 259](#)
- [Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor, page 259](#)

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
```

Example Implementing BGP Dynamic Neighbors Using Subnet Ranges

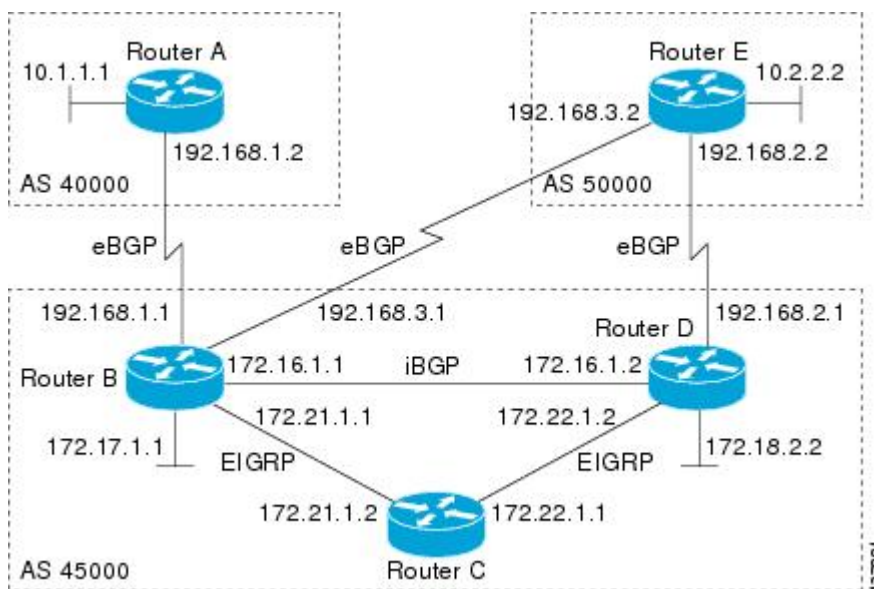
In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added 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 24 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
```

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" module.
- If you want to configure some advanced BGP features including BGP next-hop address tracking and route dampening, see the "Configuring Advanced BGP Features" module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
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
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases
Bidirectional Forwarding Detection configuration tasks	<i>Cisco IOS XE 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	Path MTU Discovery
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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17 Feature Information for Configuring BGP Neighbor Session Options Features

Feature Name	Releases	Feature Information
BGP Dynamic Neighbors	12.2(33)SXH 15.1(2)T 15.0(1)S	<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. The output for three show commands has been updated to display information about dynamic neighbors.</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, show ip bgp summary.</p>

Feature Name	Releases	Feature Information
BGP Restart Session After Max-Prefix Limit	12.0(22)S 12.2(15)T 12.2(18)S 15.0(1)S	<p>The BGP Restart Session After Max-Prefix Limit feature enhanced the capabilities of the neighbor maximum-prefix command with the introduction of the restart keyword. This enhancement allows the network operator to configure the time interval at which a peering session is reestablished by a router when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit.</p> <p>The following commands were modified by this release: neighbor maximum-prefix, show ip bgp neighbors.</p>
BGP Selective Address Tracking	12.4(4)T 12.2(31)SB 12.2(33)SRB	<p>The BGP Selective Address Tracking feature introduced the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes.</p> <p>The following commands were modified by this feature: bgp nexthop, neighbor fall-over.</p>

Feature Name	Releases	Feature Information
BGP Support for 4-Byte ASN	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 15.0(1)S	<p>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers.</p> <p>In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, and 12.2(33)SX11, the Cisco implementation of 4-byte autonomous system numbers uses <code>asplain</code> as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the <code>asplain</code> format and the <code>asdot</code> format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to <code>asdot</code> format, use the <code>bgp asnotation dot</code> command.</p> <p>In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses <code>asdot</code> as the only configuration format, regular expression match, and output display, with no <code>asplain</code> support.</p> <p>The following commands were introduced or modified by this feature: <code>bgp asnotation dot</code>, <code>bgp confederation identifier</code>, <code>bgp confederation peers</code>, all <code>clear ip bgp</code> commands that configure an autonomous system number, <code>ip as-path access-list</code>, <code>ip extcommunity-list</code>, <code>match source-protocol</code>, <code>neighbor local-as</code>, <code>neighbor remote-as</code>, <code>neighbor soo</code>, <code>redistribute (IP)</code>, <code>router bgp</code>, <code>route-target</code>, <code>set as-path</code>, <code>set extcommunity</code>, <code>set origin</code>, <code>soo</code>, all <code>show ip bgp</code> commands that display an</p>

Feature Name	Releases	Feature Information
		autonomous system number, and show ip extcommunity-list .
BGP Support for Dual AS Configuration for Network AS Migrations	12.0(27)S 12.2(25)S 12.3(11)T 12.2(33)SRA 12.2(33)SXH 15.0(1)S	<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>
BGP Support for Fast Peering Session Deactivation	12.0(29)S 12.3(14)T 12.2(33)SRA 12.2(31)SB 12.2(33)SXH 15.0(1)S	<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>

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>
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	<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>

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: 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)

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



Configuring Internal BGP Features

This module describes how to configure internal Border Gateway Protocol (BGP) features. Internal BGP (iBGP) refers to running Border Gateway Protocol (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 269](#)
- [Information About Internal BGP Features, page 269](#)
- [How to Configure Internal BGP Features, page 275](#)
- [Internal BGP Feature Configuration Examples, page 284](#)
- [Additional References, page 286](#)
- [Feature Information for Configuring Internal BGP Features, page 288](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Information About Internal BGP Features

- [BGP Routing Domain Confederation, page 269](#)
- [BGP Route Reflector, page 270](#)
- [BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer, page 273](#)
- [BGP VPLS Autodiscovery Support on Route Reflector, page 274](#)
- [BGP Route Dampening, page 274](#)

BGP Routing Domain Confederation

One way to reduce the internal BGP (iBGP) mesh is to divide an autonomous system into multiple subautonomous systems and group them into a single confederation. To the outside world, the

confederation looks like a single autonomous system. Each autonomous system is fully meshed within itself, and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have external BGP (eBGP) sessions, they exchange routing information as if they were iBGP peers. Specifically, the next hop, Multi_Exit_Discriminator (MED) attribute, and local preference information is preserved. This feature allows you to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

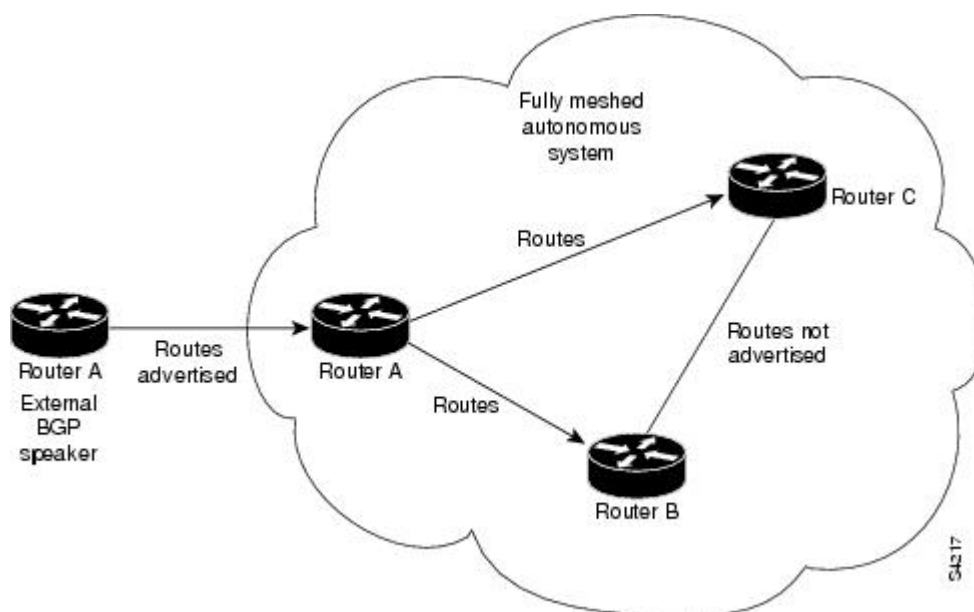
To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number.

BGP Route Reflector

BGP requires that all iBGP speakers be fully meshed. However, this requirement does not scale well when there are many iBGP speakers. Instead of configuring a confederation, another way to reduce the iBGP mesh is to configure a route reflector.

The figure below illustrates a simple iBGP configuration with three iBGP speakers (Routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both routers B and C. Routers B and C do not readvertise the iBGP learned route to other iBGP speakers because the routers do not pass on routes learned from internal neighbors to other internal neighbors, thus preventing a routing information loop.

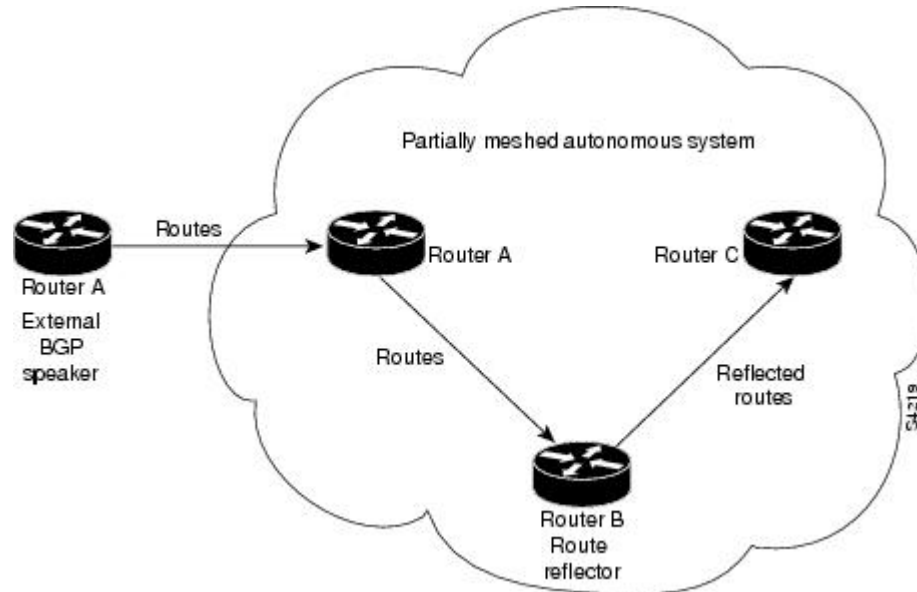
Figure 25 Three Fully Meshed iBGP Speakers



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.

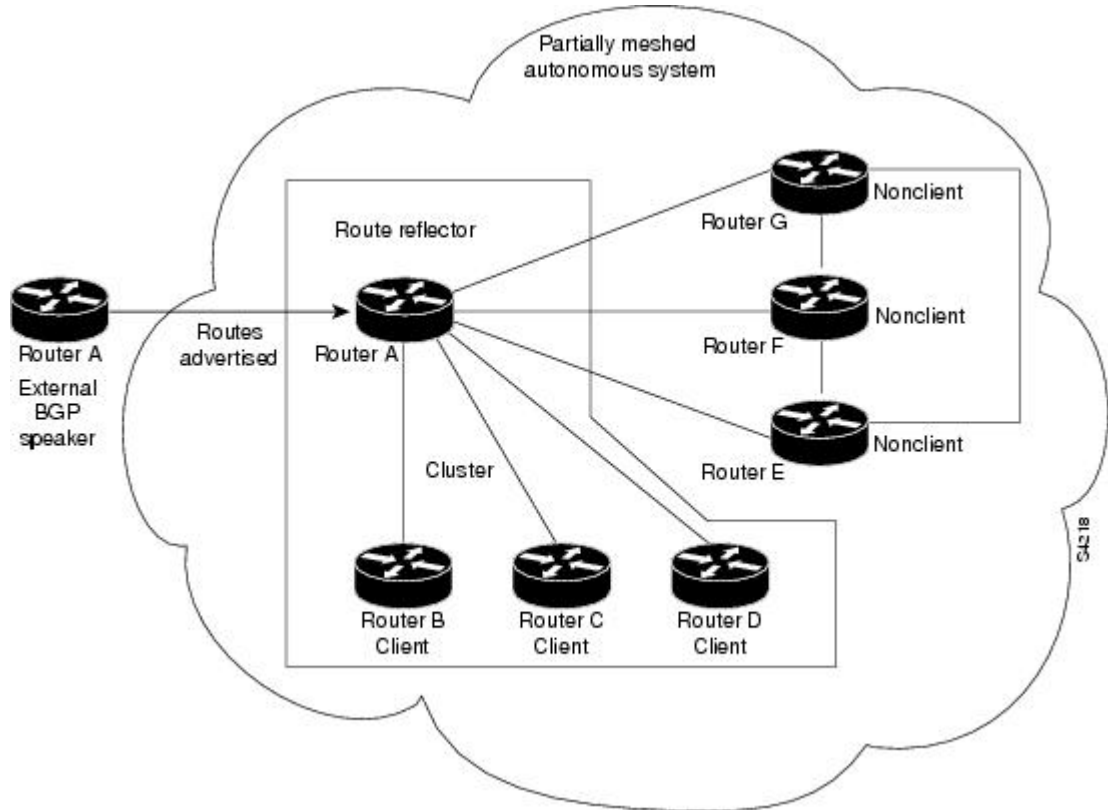
Figure 26 Simple BGP Model with a Route Reflector



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 27 More Complex BGP Route Reflector Model



When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups allowing an easy and gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All the other iBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group or nonclient group. In a simple configuration, the backbone could be divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all the route reflectors will be fully meshed). The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

Usually a cluster of clients will have a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might

have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

- [Route Reflector Mechanisms to Avoid Routing Loops, page 273](#)

Route Reflector Mechanisms to Avoid Routing Loops

As the iBGP learned routes are reflected, routing information may loop. The route reflector model has the following mechanisms to avoid routing loops:

- Originator ID is an optional, nontransitive BGP attribute. It is a 4-byte attribute created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.
- Cluster-list is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, and vice versa, it appends the local cluster ID to the cluster list. If the cluster list is empty, a new cluster list is created. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster list, the advertisement is ignored.
- The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map that is acted upon is the **set ip next-hop** clause.

BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer

The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.

The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map on a route reflector (RR) that is acted upon is the **set ip next-hop** clause. The **set ip next-hop** clause is applied to reflected routes.

Configuring an RR with an outbound route map allows a network administrator to modify the next hop attribute for a reflected route. By configuring a route map with the **set ip next-hop** clause, the administrator puts the RR into the forwarding path, and can configure iBGP multipath load sharing to achieve load balancing. That is, the RR can distribute outgoing packets among multiple egress points. See the "Configuring iBGP Multipath Load Sharing" module.



Caution

Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.

BGP VPLS Autodiscovery Support on Route Reflector

In Cisco IOS 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. For more information about VPLS Autodiscovery, see the VPLS Autodiscovery:BGP Based chapter in the *Cisco IOS MPLS Configuration Guide*.

BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.



Note

No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

- [Route Dampening Minimizes Route Flapping, page 274](#)
- [BGP Route Dampening Terms, page 274](#)

Route Dampening Minimizes Route Flapping

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

BGP Route Dampening Terms

The following terms are used when describing route dampening:

- **Flap**--A route whose availability alternates repeatedly.
- **History state**--After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- **Penalty**--Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
- **Damp state**--In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- **Suppress limit**--A route is suppressed when its penalty exceeds this limit. The default value is 2000.
- **Half-life**--Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.
- **Reuse limit**--As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
- **Maximum suppress limit**--This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevent the iBGP peers from having a higher penalty for routes external to the autonomous system.

How to Configure Internal BGP Features

- [Configuring a Routing Domain Confederation, page 275](#)
- [Configuring a Route Reflector, page 276](#)
- [Configuring a Route Reflector Using a Route Map to Set Next Hop for iBGP Peer, page 276](#)
- [Adjusting BGP Timers, page 280](#)
- [Configuring the Router to Consider a Missing MED as Worst Path, page 281](#)
- [Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths, page 281](#)
- [Configuring the Router to Use the MED to Choose a Path in a Confederation, page 281](#)
- [Enabling BGP Route Dampening, page 282](#)
- [Monitoring and Maintaining BGP Route Dampening, page 282](#)

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> [<i>as-number</i>]	Specifies the autonomous systems that belong to the confederation.

For an alternative way to reduce the iBGP mesh, see "[Configuring a Route Reflector, page 276](#)."

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 Set Next Hop for iBGP Peer

Perform this task on an RR to set a next hop for an iBGP peer. One reason to perform this task is when you want to make the RR the next hop for routes, so that you can configure iBGP load sharing. Create a route

map that sets the next hop to be the RR's address, which will be advertised to the RR clients. The route map is applied only to outbound routes from the router to which the route map is applied.


Caution

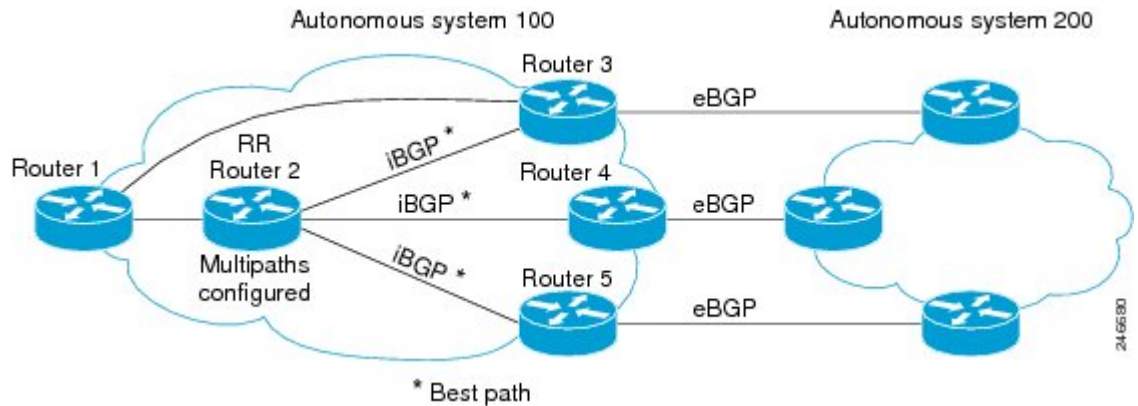
Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.


Note

Do not use the **neighbor next-hop-self** command to modify the next hop attribute for an RR. Using the **neighbor next-hop-self** command on the RR will modify next hop attributes only for non-reflected routes and not the intended routes that are being reflected from the RR clients. To modify the next hop attribute when reflecting a route, use an outbound route map.

This task configures the RR (Router 2) in the scenario illustrated in the figure below. In this case, Router 1 is the iBGP peer whose routes' next hop is being set. Without a route map, outbound routes from Router 1 would go to next hop Router 3. Instead, setting the next hop to the RR's address will cause routes from Router 1 to go to the RR, and thus allow the RR to perform load balancing among Routers 3, 4, and 5.

Figure 28 *Route Reflector Using a Route Map to Set Next Hop for an iBGP Peer*



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map *map-tag***
4. **set ip next-hop *ip-address***
5. **exit**
6. **router bgp *as-number***
7. **address-family ipv4**
8. **maximum-paths *ibgp number***
9. **neighbor *ip-address* remote-as *as-number***
10. **neighbor *ip-address* activate**
11. **neighbor *ip-address* route-reflector-client**
12. **neighbor *ip-address* route-map *map-name* out**
13. Repeat Steps 12 through 14 for the other RR clients.
14. **end**
15. **show ip bgp neighbors**

DETAILED STEPS

	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	route-map <i>map-tag</i> Example: Router(config)# route-map rr-out	Enters route map configuration mode to configure a route map. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • For this task, we want to set the next hop to be the address of the RR.

	Command or Action	Purpose
Step 5	<p>exit</p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	Exits route-map configuration mode and enters global configuration mode.
Step 6	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 100</pre>	Enters router configuration mode and creates a BGP routing process.
Step 7	<p>address-family ipv4</p> <p>Example:</p> <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	<p>maximum-paths ibgp <i>number</i></p> <p>Example:</p> <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	<p>neighbor <i>ip-address</i> remote-as <i>as-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.0.1 remote-as 100</pre>	Adds an entry to the BGP neighbor table.
Step 10	<p>neighbor <i>ip-address</i> activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.0.1 activate</pre>	Enables the exchange of information with the peer.
Step 11	<p>neighbor <i>ip-address</i> route-reflector-client</p> <p>Example:</p> <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.

Command or Action	Purpose
Step 12 <code>neighbor ip-address route-map map-name out</code> 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.
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 <code>end</code> Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.
Step 15 <code>show ip bgp neighbors</code> Example: <pre>Router# show ip bgp neighbors</pre>	(Optional) Displays information about the BGP neighbors, including their status as RR clients, and information about the route map configured.

Adjusting BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the Cisco IOS 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
<pre>Router(config-router)# timers bgp <i>keepalive</i> <i>holdtime</i></pre>	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
<pre>Router(config-router)# neighbor [<i>ip-address</i> <i>peer-group-name</i>] timers <i>keepalive holdtime</i></pre>	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 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 BGP Route Dampening

To enable BGP route dampening, use the following command in address family or router configuration mode:

Command	Purpose
Router(config-router)# bgp dampening	Enables BGP route dampening.

To change the default values of various dampening factors, use the following command in address family or router configuration mode:

Command	Purpose
Router(config-router)# bgp dampening <i>half-life reuse suppress max-suppress</i> [route-map <i>map-name</i>]	Changes the default values of route dampening factors.

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 flap-statistics	Displays BGP flap statistics for all paths.
Router# show ip bgp flap-statistics regexp <i>regexp</i>	Displays BGP flap statistics for all paths that match the regular expression.
Router# show ip bgp flap-statistics filter-list <i>access-list</i>	Displays BGP flap statistics for all paths that pass the filter.

Command	Purpose
Router# show ip bgp flap-statistics <i>ip-address mask</i>	Displays BGP flap statistics for a single entry.
Router# show ip bgp 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 dampened-paths	Displays the dampened routes, including the time remaining before they will be unsuppressed.

You can clear BGP route dampening information and unsuppress any suppressed routes by using the following command:

Command	Purpose
Router# clear ip bgp dampening [<i>ip-address network-mask</i>]	Clears route dampening information and unsuppresses the suppressed routes.

Internal BGP Feature Configuration Examples

- [Example BGP Confederation Configurations with Route Maps, page 284](#)
- [Examples BGP Confederation, page 284](#)
- [Example Route Reflector Using a Route Map to Set Next Hop for iBGP Peer, page 285](#)
- [Example BGP VPLS Autodiscovery Support on Route Reflector, page 286](#)

Example BGP Confederation Configurations with Route Maps

This section contains an example of the use of a BGP confederation configuration that includes BGP communities and route maps. For more examples of how to configure a BGP confederation, see the section [Examples BGP Confederation, page 284](#) in this chapter.

This example shows how BGP community attributes are used with a BGP confederation configuration to filter routes.

In this example, the route map named *set-community* is applied to the outbound updates to neighbor 172.16.232.50 and the local-as community attribute is used to filter the routes. The routes that pass access list 1 have the special community attribute value local-as. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers outside autonomous system 200.

```
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
!
```

Examples BGP Confederation

The following is a sample configuration that shows several peers in a confederation. The confederation consists of three internal autonomous systems with autonomous system numbers 6001, 6002, and 6003. To the BGP speakers outside the confederation, the confederation looks like a normal autonomous system with autonomous system number 500 (specified via the **bgp confederation identifier** router configuration command).

In a BGP speaker in autonomous system 6001, the **bgp confederation peers** router configuration command marks the peers from autonomous systems 6002 and 6003 as special eBGP peers. Hence peers 172.16.232.55 and 172.16.232.56 will get the local preference, next hop, and MED unmodified in the updates. The router at 10.16.69.1 is a normal eBGP speaker and the updates received by it from this peer will be just like a normal eBGP update from a peer in autonomous system 6001.

```
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 Next Hop for iBGP Peer

The following example is based on the figure above. Router 2 is the route reflector for the clients: Routers 1, 3, 4, and 5. Router 1 is connected to Router 3, but you don't want Router 1 to forward traffic destined to AS 200 to use Router 3 as the next hop (and therefore use the direct link with Router 3); you want to direct the traffic to the RR, which can load share among Routers 3, 4, and 5.

This example configures the RR, Router 2. A route map named rr-out is applied to Router 1; the route map sets the next hop to be the RR at 10.2.0.1. When Router 1 sees that the next hop is the RR address, Router 1 forwards the routes to the RR. When the RR receives packets, it will automatically load share among the iBGP paths. A maximum of five iBGP paths are allowed.

Router 2

```
route-map rr-out
  set ip next-hop 10.2.0.1
!
interface gigabitethernet 0/0
  ip address 10.2.0.1 255.255.0.0
router bgp 100
  address-family ipv4 unicast
  maximum-paths ibgp 5
  neighbor 10.1.0.1 remote-as 100
  neighbor 10.1.0.1 activate
  neighbor 10.1.0.1 route-reflector-client
  neighbor 10.1.0.1 route-map rr-out out
!
  neighbor 10.3.0.1 remote-as 100
  neighbor 10.3.0.1 activate
  neighbor 10.3.0.1 route-reflector-client
!
  neighbor 10.4.0.1 remote-as 100
```

```

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 BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge Route Reflector) is configured as a route reflector capable of reflecting VPLS prefixes. The VPLS address family is configured by **address-family l2vpn vpls** below.

```

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

The following sections provide references related to configuring internal BGP features.

Related Documents

Related Topic	Document Title
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
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>Cisco IOS IP Routing: Protocol-Independent 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
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	A Border Gateway Protocol 4 (BGP-4)
RFC 4893	BGP Support for Four-octet AS Number Space
RFC 5396	Textual Representation of Autonomous system (AS) Numbers
RFC 5398	Autonomous System (AS) Number Reservation for Documentation Use

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/techsupport</p>

Feature Information for Configuring Internal BGP Features

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

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

Table 18 Feature Information for Configuring Internal BGP Features

Feature Name	Releases	Feature Configuration Information
Configuring internal BGP features	10.3 12.0(32)S12 12.0(7)T 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 dampened-paths • show ip bgp flap-statistics • timers bgp
BGP Outbound Route Map on Route Reflector to Set IP Next Hop	12.0(22)S 12.0(16)ST 12.2 12.2(14)S 15.0(1)S	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.
BGP VPLS Autodiscovery Support on Route Reflector	12.2(33)SRE	This feature was introduced on the Cisco 7600 and Cisco 7200 series routers.

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: 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)

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



Configuring Advanced BGP Features

This module describes configuration tasks for various advanced Border Gateway Protocol (BGP) features. BGP is an interdomain routing protocol designed to provide loop-free routing between organizations. This module contains tasks to configure BGP next-hop address tracking, BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability, route dampening, Bidirectional Forwarding Detection (BFD) support for BGP, BGP MIB support and BGP support for Multi-Topology Routing (MTR).

- [Finding Feature Information, page 291](#)
- [Prerequisites for Configuring Advanced BGP Features, page 291](#)
- [Restrictions for Configuring Advanced BGP Features, page 291](#)
- [Information About Configuring Advanced BGP Features, page 292](#)
- [How to Configure Advanced BGP Features, page 302](#)
- [Where to Go Next, page 341](#)
- [Additional References, page 341](#)
- [Feature Information for Configuring Advanced BGP Features, page 342](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Configuring Advanced BGP Features

Before configuring advanced BGP features you should be familiar with the “Cisco BGP Overview” module and the “Configuring a Basic BGP Network” module.

Restrictions for Configuring Advanced BGP Features

- A router that runs Cisco IOS 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.
- Multicast BGP peer support is not available in Cisco IOS software after Release 12.2(33)SRA.

Information About Configuring Advanced BGP Features

- [BGP Version 4](#), page 292
- [BGP Support for Next-Hop Address Tracking](#), page 292
- [BGP Nonstop Forwarding Awareness](#), page 293
- [BGP Route Dampening](#), page 297
- [BFD for BGP](#), page 298
- [BGP MIB Support](#), page 298
- [BGP Support for MTR](#), page 300

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS). For more details about configuring a basic BGP network, see the “Configuring a Basic BGP Network” module.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. For more details about connecting to external BGP peers, see the “Connecting to a Service Provider Using External BGP” 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 Support for Next-Hop Address Tracking

To configure BGP next-hop address tracking you should understand the following concepts:

- [BGP Next-Hop Address Tracking](#), page 292
- [Default BGP Scanner Behavior](#), page 293
- [Selective BGP Next-Hop Route Filtering](#), page 293
- [BGP Next_Hop Attribute](#), page 293

BGP Next-Hop Address Tracking

The BGP next-hop address tracking feature is enabled by default when a supporting Cisco software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best-path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

Default BGP Scanner Behavior

BGP monitors the next hop of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. By default, the BGP scanner is used to poll the RIB for this information every 60 seconds. During the 60 second time period between scan cycles, Interior Gateway Protocol (IGP) instability or other network failures can cause black holes and routing loops to temporarily form.

Selective BGP Next-Hop Route Filtering

In Cisco IOS Release 12.4(4)T, 12.2(33)SRB, and later releases, 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**

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.

BGP Next_Hop Attribute

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

BGP Nonstop Forwarding Awareness

To configure BGP Nonstop Forwarding (NSF) awareness you should understand the following concepts:

- [Cisco NSF Routing and Forwarding Operation, page 294](#)
- [Cisco Express Forwarding for NSF, page 294](#)
- [BGP Graceful Restart for NSF, page 295](#)
- [BGP NSF Awareness, page 295](#)
- [BGP Graceful Restart per Neighbor, page 296](#)
- [BGP Peer Session Templates, page 296](#)

Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, EIGRP, OSPF, and IS-IS have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

In this document, a networking device is said to be NSF-aware if it is running NSF-compatible software. A device is said to be NSF-capable if it has been configured to support NSF; therefore, it would rebuild routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the line cards with the new FIB information.



Note

Currently, EIGRP supports only NSF awareness.

Cisco Express Forwarding for NSF

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by CEF. CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version (epoch) number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

**Note**

For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

BGP Graceful Restart for NSF

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable or NSF-aware router has graceful restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peer(s) (NSF-aware peers) need to exchange the graceful restart capability in their OPEN messages, at the time of session establishment. If both the peers do not exchange the graceful restart capability, the session will not be graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the NSF-capable device. This functionality will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

BGP NSF Awareness

BGP support for NSF requires that neighbor routers are NSF-aware or NSF-capable. NSF awareness in BGP is also enabled by the graceful restart mechanism. A router that is NSF-aware functions like a router that is NSF-capable with one exception: an NSF-aware router is incapable of performing an SSO operation. However, a router that is NSF-aware is capable of maintaining a peering relationship with a NSF-capable neighbor during a NSF SSO operation, as well as holding routes for this neighbor during the SSO operation.

The BGP Nonstop Forwarding Awareness feature provides an NSF-aware router with the capability to detect a neighbor that is undergoing an SSO operation, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes. The deployment of BGP NSF awareness can minimize the effects of Route Processor (RP) failure conditions and improve the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

NSF awareness for BGP is not enabled by default. The **bgp graceful-restart** command is used to globally enable NSF awareness on a router that is running BGP. NSF-aware operations are also transparent to the network operator and BGP peers that do not support NSF capabilities.

**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.

BGP Graceful Restart per Neighbor

In Cisco IOS Releases 12.2(33)SRC, 12.2(33)SB (on platforms including the Cisco 10000 series routers), 15.0(1)M, and later releases, the ability to enable or disable BGP graceful restart for every individual BGP neighbor was introduced. Three new methods of configuring BGP graceful restart for BGP peers, in addition to the existing global BGP graceful restart configuration, are now available. Graceful restart can be enabled or disabled for a BGP peer or a BGP peer group using the **neighbor ha-mode graceful-restart** command, or a BGP peer can inherit a graceful restart configuration from a BGP peer-session template using the **ha-mode graceful-restart** command.

Although BGP graceful restart is disabled by default, the existing global command enables graceful restart for all BGP neighbors regardless of their capabilities. The ability to enable or disable BGP graceful restart for individual BGP neighbors provides a greater level of control for a network administrator.

When the BGP graceful restart capability is configured for an individual neighbor, each method of configuring graceful restart has the same priority, and the last configuration instance is applied to the neighbor. For example, if global graceful restart is enabled for all BGP neighbors but an individual neighbor is subsequently configured as a member of a peer group for which the graceful restart is disabled, graceful restart is disabled for that neighbor.

The configuration of the restart and stale-path timers is available only with the global **bgp graceful-restart** command, but the default values are set when the **neighbor ha-mode graceful-restart** or **ha-mode graceful-restart** commands are configured. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

BGP Peer Session Templates

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

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

Peer session templates support direct and indirect inheritance. A BGP neighbor can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template. A BGP neighbor can directly inherit only one session template and can indirectly inherit up to seven additional peer session templates.

Peer session templates support inheritance. A directly applied peer session template can directly or indirectly inherit configurations from up to seven peer session templates. So, a total of eight peer session templates can be applied to a neighbor or neighbor group.

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

For more details about BGP peer session templates, see the section “Configuring a Basic BGP Network.”

To use a BGP peer session template to enable or disable BGP graceful restart, see the section “Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates.”

BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.



Note

No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

Minimizing Flapping

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

Understanding Route Dampening Terms

The following terms are used when describing route dampening:

- Flap—A route whose availability alternates repeatedly.
- History state—After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- Penalty—Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored

in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.

- Damp state—In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- Suppress limit—A route is suppressed when its penalty exceeds this limit. The default value is 2000.
- Half-life—Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.
- Reuse limit—As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
- Maximum suppress limit—This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevent the iBGP peers from having a higher penalty for routes external to the autonomous system.

BFD for BGP

Bidirectional Forwarding Detection (BFD) support for BGP was introduced in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.0(32)S, 12.2(33)SRA, 12.2(33)SXH, 12.2(33)SB, and later releases. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a marked decrease in reconvergence time.

One caveat exists for BFD; BFD and BGP graceful restart capability cannot both be configured on a router running BGP. If an interface goes down, BFD detects the failure and indicates that the interface cannot be used for traffic forwarding and the BGP session goes down, but graceful restart still allows traffic forwarding on platforms that support NSF even though the BGP session is down, allowing traffic forwarding using the interface that is down. Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing.

For more details about BFD, see the "Bidirectional Forwarding Detection" module of the *Cisco IOS IP Routing: BFD Configuration Guide*.

BGP MIB Support

The Management Information Base (MIB) to support BGP is the CISCO-BGP4-MIB. In Cisco IOS Release 12.0(26)S, 12.3(7)T, 12.2(25)S, 12.2(33)SRA, 12.2(33)SXH, and later releases, the BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following sections describe the objects and notifications (traps) that are supported:

BGP FSM Transition Change Support

The *cbgpRouteTable* supports BGP Finite State Machine (FSM) transition state changes.

The *cbgpFsmStateChange* object allows you to configure SNMP notifications (traps) for all FSM transition state changes. This notification contains the following MIB objects:

- *bgpPeerLastError*
- *bgpPeerState*
- *cbgpPeerLastErrorTxt*
- *cbgpPeerPrevState*

The *cbgpBackwardTransition* object supports all BGP FSM transition state changes. This object is sent each time the FSM moves to either a higher or lower numbered state. This notification contains the following MIB objects:

- *bgpPeerLastError*
- *bgpPeerState*
- *cbgpPeerLastErrorTxt*
- *cbgpPeerPrevState*

The **snmp-server enable bgp traps** command allows you to enable the traps individually or together with the existing FSM backward transition and established state traps as defined in [RFC 1657](#).

BGP Route Received Route Support

The *cbgpRouteTable* object supports the total number of routes received by a BGP neighbor. The following MIB object is used to query the CISCO-BGP4-MIB for routes that are learned from individual BGP peers:

- *cbgpPeerAddrFamilyPrefixTable*

Routes are indexed by the address-family identifier (AFI) or subaddress-family identifier (SAFI). The prefix information displayed in this table can also be viewed in the output of the **show ip bgp** command.

BGP Prefix Threshold Notification Support

The *cbgpPrefixMaxThresholdExceed* and *cbgpPrfexixMaxThresholdClear* 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 *cbgpPrfexixMaxThresholdClear* 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*

BGP Support for MTR

BGP support for MTR was introduced in Cisco IOS Release 12.2(33)SRB. For more details, see the "Multi-Topology Routing" documentation. Before using BGP to support MTR, you should be familiar with the following concepts:

- [BGP Network Scope, page 300](#)
- [MTR CLI Hierarchy Under BGP, page 301](#)
- [BGP Sessions for Class-Specific Topologies, page 301](#)
- [Topology Translation Using BGP, page 302](#)
- [Topology Import Using BGP, page 302](#)

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 Advanced BGP Features

- [Configuring BGP Next-Hop Address Tracking, page 302](#)
- [Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart, page 309](#)
- [Configuring BGP Route Dampening, page 324](#)
- [Decreasing BGP Convergence Time Using BFD, page 327](#)
- [Enabling BGP MIB Support, page 331](#)
- [Configuring BGP Support for MTR, page 332](#)

Configuring BGP Next-Hop Address Tracking

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 configuring route dampening, see “Configuring BGP Route Dampening.”

- [Disabling BGP Next-Hop Address Tracking, page 302](#)
- [Adjusting the Delay Interval for BGP Next-Hop Address Tracking, page 304](#)
- [Configuring BGP Selective Next-Hop Route Filtering, page 305](#)

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 reenab 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: 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 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: 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.

Command or Action	Purpose
Step 5 <code>no bgp nexthop trigger enable</code> Example: <pre>Router(config-router-af)# no bgp nexthop trigger enable</pre>	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 <code>end</code> Example: <pre>Router(config-router-af)# end</pre>	Exits address-family configuration mode, and enters Privileged EXEC mode.

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

- `enable`
- `configure terminal`
- `router bgp autonomous-system-number`
- `address-family ipv4 [[mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name] | vpn4 [unicast]]`
- `bgp nexthop trigger delay delay-timer`
- `end`

DETAILED STEPS

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

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 64512</pre>	Enters router configuration mode to create or configure a BGP routing process.
<p>Step 4 <code>address-family ipv4 [[<i>mdt</i> <i>multicast</i> <i>tunnel</i> <i>unicast</i> [<i>vrf vrf-name</i>] <i>vrf vrf-name</i>] <i>vpn4</i> [<i>unicast</i>]]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre>	<p>Enter address family configuration mode to configure BGP peers to accept address family-specific configurations.</p> <ul style="list-style-type: none"> The example creates an IPv4 unicast address family session.
<p>Step 5 <code>bgp nexthop trigger delay <i>delay-timer</i></code></p> <p>Example:</p> <pre>Router(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.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address-family configuration mode, and enters privileged EXEC mode.

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](#).

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: 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 and creates a BGP routing process.

Command or Action	Purpose
<p>Step 4 <code>address-family ipv4 [unicast multicast] vrf vrf-name]</code></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>Step 5 <code>bgp nexthop route-map map-name</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 6 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	<p>Exits address family configuration mode and enters router configuration mode.</p>
<p>Step 7 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>
<p>Step 8 <code>ip prefix-list list-name [seq seq-value] {deny network / length permit network / length}[ge ge-value] [le le-value]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 9 <code>route-map map-name [permit deny] [sequence-number]</code></p> <p>Example:</p> <pre>Router(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.

Command or Action	Purpose
<p>Step 10 <code>match ip address prefix-list prefix-list-name [prefix-list-name...]</code></p> <p>Example:</p> <pre>Router(config-route-map)# match ip address prefix-list FILTER25</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>
<p>Step 11 <code>exit</code></p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and enters global configuration mode.</p>
<p>Step 12 <code>route-map map-name [permit deny] [sequence-number]</code></p> <p>Example:</p> <pre>Router(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.
<p>Step 13 <code>end</code></p> <p>Example:</p> <pre>Router(config-route-map)# end</pre>	<p>Exits route map configuration mode and enters privileged EXEC mode.</p>
<p>Step 14 <code>show ip bgp [network] [network-mask]</code></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 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
```

Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart

The tasks in this section show how to configure BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability. The first task enables BGP NSF globally for all BGP neighbors and suggests a few troubleshooting options. The second task describes how to adjust the BGP graceful restart timers although the default settings are optimal for most network deployments. The next three tasks demonstrate how to enable or disable BGP graceful restart for individual BGP neighbors including peer session templates and peer groups. The final task verifies the local and peer router configuration of BGP NSF.

- [Enabling BGP Global NSF Awareness Using BGP Graceful Restart, page 309](#)
- [Configuring BGP NSF Awareness Timers, page 311](#)
- [Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates, page 313](#)
- [Enabling BGP Graceful Restart for an Individual BGP Neighbor, page 318](#)
- [Disabling BGP Graceful Restart for a BGP Peer Group, page 320](#)
- [Verifying the Configuration of BGP Nonstop Forwarding Awareness, page 323](#)

Enabling BGP Global NSF Awareness Using BGP Graceful Restart

Perform this task to enable BGP NSF awareness globally for all BGP neighbors. BGP NSF awareness is part of the graceful restart mechanism and BGP NSF awareness is enabled by issuing the **bgp graceful-restart** command in router configuration mode. BGP NSF awareness allows NSF-aware routers to support NSF-capable routers during an SSO operation. NSF-awareness is not enabled by default and should be configured on all neighbors that participate in BGP NSF.

**Note**

The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

**Note**

Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see the section “BFD for BGP.”

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart** [**restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 <code>enable</code> Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>router bgp <i>autonomous-system-number</i></code> Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4 <code>bgp graceful-restart [<i>restart-time seconds</i>]</code> <code>[<i>stalepath-time seconds</i>]</code> Example: <pre>Router(config-router)# bgp graceful-restart</pre>	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 <code>end</code> Example: <pre>Router(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

- [Troubleshooting Tips, page 310](#)
- [What to Do Next, page 311](#)

Troubleshooting Tips

To troubleshoot the NSF feature, use the following commands in privileged EXEC mode, as needed:

- debug ip bgp** Displays open messages that advertise the graceful restart capability.
- debug ip bgp event** Displays graceful restart timer events, such as the restart timer and the stalepath timer.
- debug ip bgp updates** Displays sent and received EOR messages. The EOR message is used by the NSF-aware router to start the stalepath timer, if configured.

- **show ip bgp** Displays entries in the BGP routing table. The output from this command will display routes that are marked as stale by displaying the letter “S” next to each stale route.
- **show ip bgp neighbor** Displays information about the TCP and BGP connections to neighbor devices. When enabled, the graceful restart capability is displayed in the output of this command.

What to Do Next

If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the “Configuring a Basic BGP Network” module.

Configuring BGP NSF Awareness Timers

Perform this task to adjust the BGP graceful restart timers. There are two BGP graceful restart timers that can be configured. The optional **restart-time** keyword and *seconds* argument determine how long peer routers will wait to delete stale routes before a BGP open message is received. The default value is 120 seconds. The optional **stalepath-time** keyword and *seconds* argument determine how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. The default value is 360 seconds.



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
<p>Step 1 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.

Command or Action	Purpose
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>router bgp autonomous-system-number</code> Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4 <code>bgp graceful-restart [restart-time seconds]</code> Example: <pre>Router(config-router)# bgp graceful-restart restart-time 130</pre>	<p>Enables the BGP graceful restart capability and BGP NSF awareness.</p> <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 configurable range is from 1 to 3600 seconds. <p>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>.</p>
Step 5 <code>bgp graceful-restart [stalepath-time seconds]</code> Example: <pre>Router(config-router)# bgp graceful-restart stalepath-time 350</pre>	<p>Enables the BGP graceful restart capability and BGP NSF awareness.</p> <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 configurable range is from 1 to 3600 seconds. <p>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>.</p>
Step 6 <code>end</code> Example: <pre>Router(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

- [What to Do Next, page 312](#)

What to Do Next

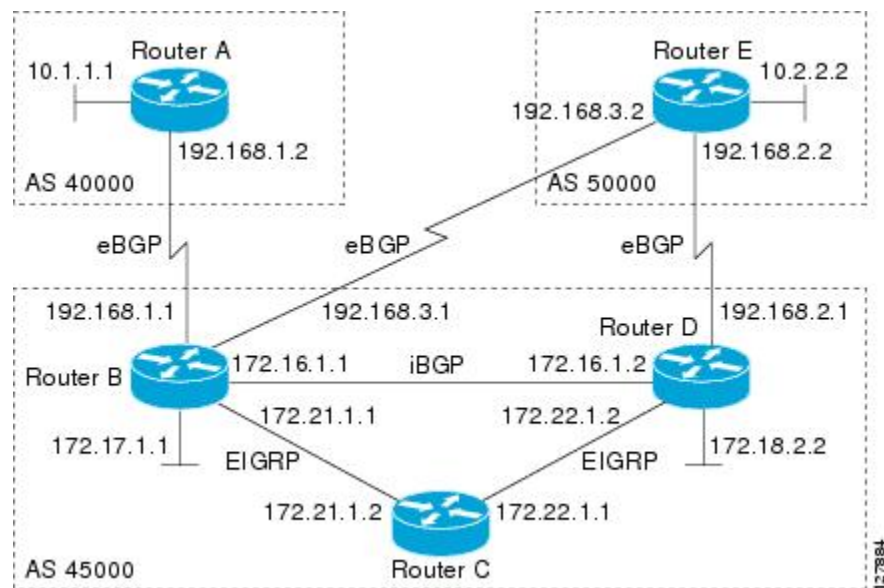
If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset the peer sessions by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the “Configuring a Basic BGP Network” module.

Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

Perform this task to enable and disable BGP graceful restart for BGP neighbors using peer session templates. In this task, a BGP peer session template is created, and BGP graceful restart is enabled. A second peer session template is created, and this template is configured to disable BGP graceful restart.

In this example, the configuration is performed at Router B in the figure below and two external BGP neighbors--at Router A and Router E in the figure below--are identified. The first BGP peer at Router A is configured to inherit the first peer session template that enables BGP graceful restart, whereas the second BGP peer at Router E inherits the second template that disables BGP graceful restart. Using the optional **show ip bgp neighbors** command, the status of the BGP graceful restart capability is verified for each BGP neighbor configured in this task.

Figure 29 Network Topology Showing BGP Neighbors



The restart and stale-path timers can be modified only using the global **bgp graceful-restart** command as shown in the [Configuring BGP NSF Awareness Timers](#), page 311. The restart and stale-path timers are set to the default values when BGP graceful restart is enabled for BGP neighbors using peer session templates.

This task requires a Cisco IOS Release 12.2(33)SRC, or 12.2(33)SB.



Note

A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode graceful-restart** [**disable**]
6. **exit-peer-session**
7. **template peer-session** *session-template-name*
8. **ha-mode graceful-restart** [**disable**]
9. **exit-peer-session**
10. **bgp log-neighbor-changes**
11. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
12. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
15. **end**
16. **show ip bgp template peer-session** [*session-template-number*]
17. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

DETAILED STEPS

	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 and creates a BGP routing process.

	Command or Action	Purpose
Step 4	<p>template peer-session <i>session-template-name</i></p> <p>Example:</p> <pre>Router(config-router)# template peer-session S1</pre>	<p>Enters session-template configuration mode and creates a peer session template.</p> <ul style="list-style-type: none"> In this example, a peer session template named S1 is created.
Step 5	<p>ha-mode graceful-restart [disable]</p> <p>Example:</p> <pre>Router(config-router-stmp)# ha-mode graceful-restart</pre>	<p>Enables the BGP graceful restart capability and BGP NSF awareness.</p> <ul style="list-style-type: none"> Use the disable keyword to disable BGP graceful restart capability. If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. In this example, the BGP graceful restart capability is enabled for the peer session template named S1.
Step 6	<p>exit-peer-session</p> <p>Example:</p> <pre>Router(config-router-stmp)# exit-peer-session</pre>	<p>Exits session-template configuration mode and returns to router configuration mode.</p>
Step 7	<p>template peer-session <i>session-template-name</i></p> <p>Example:</p> <pre>Router(config-router)# template peer-session S2</pre>	<p>Enters session-template configuration mode and creates a peer session template.</p> <ul style="list-style-type: none"> In this example, a peer session template named S2 is created.
Step 8	<p>ha-mode graceful-restart [disable]</p> <p>Example:</p> <pre>Router(config-router-stmp)# ha-mode graceful-restart disable</pre>	<p>Enables the BGP graceful restart capability and BGP NSF awareness.</p> <ul style="list-style-type: none"> Use the disable keyword to disable BGP graceful restart capability. If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. In this example, the BGP graceful restart capability is disabled for the peer session template named S2.
Step 9	<p>exit-peer-session</p> <p>Example:</p> <pre>Router(config-router-stmp)# exit-peer-session</pre>	<p>Exits session-template configuration mode and returns to router configuration mode.</p>

Command or Action	Purpose
<p>Step 10 <code>bgp log-neighbor-changes</code></p> <p>Example:</p> <pre>Router(config-router)# bgp log-neighbor-changes</pre>	<p>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.
<p>Step 11 <code>neighbor ip-address remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 remote-as 40000</pre>	<p>Configures peering with a BGP neighbor in the specified autonomous system.</p> <ul style="list-style-type: none"> In this example, the BGP peer at 192.168.1.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
<p>Step 12 <code>neighbor ip-address inherit peer-session session-template-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.2 inherit peer-session S1</pre>	<p>Inherits a peer session template.</p> <ul style="list-style-type: none"> In this example, the peer session template named S1 is inherited, and the neighbor inherits the enabling of BGP graceful restart.
<p>Step 13 <code>neighbor ip-address remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.2 remote-as 50000</pre>	<p>Configures peering with a BGP neighbor in the specified autonomous system.</p> <ul style="list-style-type: none"> In this example, the BGP peer at 192.168.3.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
<p>Step 14 <code>neighbor ip-address inherit peer-session session-template-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.3.2 inherit peer-session S2</pre>	<p>Inherits a peer session-template.</p> <ul style="list-style-type: none"> In this example, the peer session template named S2 is inherited, and the neighbor inherits the disabling of BGP graceful restart.
<p>Step 15 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 16 <code>show ip bgp template peer-session [session-template-number]</code></p> <p>Example:</p> <pre>Router# show ip bgp template peer-session</pre>	<p>(Optional) 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 <i>session-template-name</i> argument. This command also supports all standard output modifiers.
<p>Step 17 <code>show ip bgp neighbors [ip-address [received-routes routes advertised-routes paths [regexp] dampened-routes flap-statistics] received prefix-filter policy[detail]]]</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors 192.168.1.2</pre>	<p>(Optional) Displays information about TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. In this example, the output is filtered to display information about the BGP peer at 192.168.1.2.

Examples

The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 192.168.1.2 (Router A in the figure above). Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can only be set using the global `bgp graceful-restart` command.

```
Router# show ip bgp neighbors 192.168.1.2
BGP neighbor is 192.168.1.2, remote AS 40000, external link
Inherits from template S1 for session parameters
BGP version 4, remote router ID 192.168.1.2
BGP state = Established, up for 00:02:11
Last read 00:00:23, last write 00:00:27, hold time is 180, keepalive intervals
Neighbor sessions:
  1 active, is multisession capable
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Address family IPv4 Unicast: advertised and received
  Graceful Restart Capability: advertised
  Multisession Capability: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.1.2
Connections established 1; dropped 0
Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

The following example shows partial output from the `show ip bgp neighbors` command for the BGP peer at 192.168.3.2 (Router E in the figure above). Graceful restart is shown as disabled.

```
Router# show ip bgp neighbors 192.168.3.2
BGP neighbor is 192.168.3.2, remote AS 50000, external link
Inherits from template S2 for session parameters
BGP version 4, remote router ID 192.168.3.2
BGP state = Established, up for 00:01:41
Last read 00:00:45, last write 00:00:45, hold time is 180, keepalive intervals
Neighbor sessions:
```

```

    1 active, is multisesion capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  !
Address tracking is enabled, the RIB does have a route to 192.168.3.2
Connections established 1; dropped 0
Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0

```

Enabling BGP Graceful Restart for an Individual BGP Neighbor

Perform this task on Router B in the figure above to enable BGP graceful restart on the internal BGP peer at Router C in the figure above. Under address family IPv4, the neighbor at Router C is identified, and BGP graceful restart is enabled for the neighbor at Router C with the IP address 172.21.1.2. To verify that BGP graceful restart is enabled, the optional **show ip bgp neighbors** command is used.

This task requires a Cisco IOS Release 12.2(33)SRC, 12.2(33)SB, or 15.0(1)M.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*]
5. **neighbor ip-address remote-as** *autonomous-system-number*
6. **neighbor ip-address activate**
7. **neighbor ip-address ha-mode graceful-restart** [**disable**]
8. **end**
9. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**] **received prefix-filter** | **policy**[**detail**]]]

DETAILED STEPS

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.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode and creates a BGP routing process.</p>
<p>Step 4 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i></code></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>Step 5 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.21.1.2 remote-as 45000</pre>	<p>Configures peering with a BGP neighbor in the specified autonomous system.</p> <ul style="list-style-type: none"> In this example, the BGP peer at 172.21.1.2 is an internal BGP peer because it has the same autonomous system number as the router where the BGP configuration is being entered (see Step 3).
<p>Step 6 <code>neighbor <i>ip-address</i> activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.21.1.2 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 address family with the local router.</p> <ul style="list-style-type: none"> In this example, the internal BGP peer at 172.21.1.2 is activated.
<p>Step 7 <code>neighbor <i>ip-address</i> ha-mode graceful-restart [disable]</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.21.1.2 ha-mode graceful-restart</pre>	<p>Enables the BGP graceful restart capability for a BGP neighbor.</p> <ul style="list-style-type: none"> Use the disable keyword to disable BGP graceful restart capability. If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. In this example, the BGP graceful restart capability is enabled for the neighbor at 172.21.1.2.
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Command or Action	Purpose
Step 9 <code>show ip bgp neighbors [ip-address [received-routes routes advertised-routes paths [regexp] dampened-routes flap-statistics] received prefix-filter policy[detail]]]</code>	(Optional) Displays information about TCP and BGP connections to neighbors. <ul style="list-style-type: none"> • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 172.21.1.2.
Example:	
<pre>Router# show ip bgp neighbors 172.21.1.2</pre>	

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.21.1.2. Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

```
Router# show ip bgp neighbors 172.21.1.2
BGP neighbor is 172.21.1.2, remote AS 45000, internal link
  BGP version 4, remote router ID 172.22.1.1
  BGP state = Established, up for 00:01:01
  Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
  !
  Address tracking is enabled, the RIB does have a route to 172.21.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Disabling BGP Graceful Restart for a BGP Peer Group

Perform this task to disable BGP graceful restart for a BGP peer group. In this task, a BGP peer group is created and graceful restart is disabled for the peer group. A BGP neighbor, 172.16.1.2 at Router D in the figure above, is then identified and added as a peer group member and inherits the configuration associated with the peer group, which, in this example, disables BGP graceful restart.

This task requires a Cisco IOS Release 12.2(33)SRC, 12.2(33)SB, or 15.0(1)M.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number*
7. **neighbor** *peer-group-name* **ha-mode graceful-restart** [**disable**]
8. **neighbor** *ip-address* **peer-group** *peer-group-name*
9. **end**
10. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*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.
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	address-family ipv4 [unicast multicast] vrf <i>vrf-name</i> Example: Router(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.

	Command or Action	Purpose
Step 5	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor PG1 peer-group</pre>	<p>Creates a BGP peer group.</p> <ul style="list-style-type: none"> In this example, the peer group named PG1 is created.
Step 6	<p>neighbor <i>peer-group-name</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor PG1 remote-as 45000</pre>	<p>Configures peering with a BGP peer group in the specified autonomous system.</p> <ul style="list-style-type: none"> In this example, the BGP peer group named PG1 is added to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 7	<p>neighbor <i>peer-group-name</i> ha-mode graceful-restart [disable]</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor PG1 ha-mode graceful-restart disable</pre>	<p>Enables the BGP graceful restart capability for a BGP neighbor.</p> <ul style="list-style-type: none"> Use the disable keyword to disable BGP graceful restart capability. If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. In this example, the BGP graceful restart capability is disabled for the BGP peer group named PG1.
Step 8	<p>neighbor <i>ip-address</i> peer-group <i>peer-group-name</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.2 peer-group PG1</pre>	<p>Assigns the IP address of a BGP neighbor to a peer group.</p> <ul style="list-style-type: none"> In this example, the BGP neighbor peer at 172.16.1.2 is configured as a member of the peer group named PG1.
Step 9	<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>
Step 10	<p>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]]]</p> <p>Example:</p> <pre>Router# show ip bgp neighbors 172.16.1.2</pre>	<p>(Optional) Displays information about TCP and BGP connections to neighbors.</p> <ul style="list-style-type: none"> In this example, the output is filtered to display information about the BGP peer at 172.16.1.2 and the "Graceful-Restart is disabled" line shows that the graceful restart capability is disabled for this neighbor.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.16.1.2. Graceful restart is shown as disabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

```
Router# show ip bgp neighbors 172.16.1.2
BGP neighbor is 172.16.1.2, remote AS 45000, internal link
Member of peer-group PGI for session parameters
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
Neighbor sessions:
  0 active, is multisession capable
!
Address tracking is enabled, the RIB does have a route to 172.16.1.2
Connections established 0; dropped 0
Last reset never
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
```

Verifying the Configuration of BGP Nonstop Forwarding Awareness

Use the following steps to verify the local configuration of BGP NSF awareness on a router and to verify the configuration of NSF awareness on peer routers in a BGP network.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]
3. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*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** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]

Displays information about TCP and BGP connections to neighbors. "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. In Cisco IOS 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 multiseession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multiseession 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

```

Configuring BGP Route Dampening

The tasks in this section configure and monitor BGP route dampening. Route dampening is designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

- [Enabling and Configuring BGP Route Dampening, page 324](#)
- [Monitoring and Maintaining BGP Route Dampening, page 326](#)

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
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router bgp <i>as-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>
<p>Step 4 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.

Command or Action	Purpose
<p>Step 5 <code>bgp dampening</code> [<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.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and enters privileged EXEC mode.</p>

Monitoring and Maintaining BGP Route Dampening

Perform the steps in this task as required to monitor and maintain BGP route dampening.

SUMMARY STEPS

- `enable`
- `show ip bgp flap-statistics` [**regex** *regex* | **filter-list** *access-list* | *ip-address mask* [**longer-prefix**]]
- `clear ip bgp flap-statistics` [*neighbor-address* [*ipv4-mask*]] [**regex** *regex* | **filter-list** *extcom-number*]
- `show ip bgp dampened-paths`
- `clear ip bgp` [**ipv4** { **multicast** | **unicast** } | **ipv6** { **multicast** | **unicast** } | **vpn4 unicast**] **dampening** [*neighbor-address*] [*ipv4-mask*]

DETAILED STEPS

-
- Step 1** `enable`
Enables privileged EXEC mode. Enter your password if prompted.
- Example:**
- ```
Router> enable
```
- Step 2** `show ip bgp flap-statistics` [**regex** *regex* | **filter-list** *access-list* | *ip-address mask* [**longer-prefix**]]  
Use this command to monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life.
- Example:**
- ```
Router# show ip bgp flap-statistics
BGP table version is 10, local router ID is 172.17.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          From            Flaps  Duration Reuse    Path
*d 10.0.0.0        172.17.232.177  4      00:13:31 00:18:10 100
*d 10.2.0.0        172.17.232.177  4      00:02:45 00:28:20 100
```

Step 3 **clear ip bgp flap-statistics** [*neighbor-address* [*ipv4-mask*]] [**regexp** *regexp* | **filter-list** *extcom-number*]

Use this command to clear the accumulated penalty for routes that are received on a router that has BGP dampening enabled. If no arguments or keywords are specified, flap statistics are cleared for all routes. Flap statistics are also cleared when the peer is stable for the half-life time period. After the BGP flap statistics are cleared, the route is less likely to be dampened.

Example:

```
Router# clear ip bgp flap-statistics 172.17.232.177
```

Step 4 **show ip bgp dampened-paths**

Use this command to monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life.

Example:

```
Router# show ip bgp dampened-paths
BGP table version is 10, local router ID is 172.29.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          From            Reuse    Path
*d 10.0.0.0        172.16.232.177  00:18:4 100 ?
*d 10.2.0.0        172.16.232.177  00:28:5 100 ?
```

Step 5 **clear ip bgp** [**ipv4** {**multicast** | **unicast**} | **ipv6**{**multicast** | **unicast**} | **vpnv4 unicast**] **dampening** [*neighbor-address*] [*ipv4-mask*]

Use this command to clear stored route dampening information. If no keywords or arguments are entered, route dampening information for the entire routing table is cleared. The following example clears route dampening information for VPNv4 address family prefixes from network 192.168.10.0/24, and unsuppresses its suppressed routes.

Example:

```
Router# clear ip bgp vpnv4 unicast dampening 192.168.10.0 255.255.255.0
```

Decreasing BGP Convergence Time Using BFD

BFD support for BGP was introduced in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.2(33)SRA, 12.2(33)SXH, 12.2(33)SB, and later releases. 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 BGP Neighbor Session Options" chapter, the section "Configuring BFD for BGP IPv6 Neighbors."

- [Prerequisites, page 328](#)
- [Restrictions, page 328](#)
- [Configuring BFD Session Parameters on the Interface, page 328](#)
- [Configuring BFD Support for BGP, page 329](#)
- [Monitoring and Troubleshooting BFD for Cisco 7600 Series Routers, page 331](#)

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 current Cisco implementation of BFD support for BGP in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.2(33)SRA, 12.2(33)SXH, and 12.2(33)SB, BFD is supported only for IPv4 networks, and only asynchronous mode is supported. In asynchronous mode, either BFD peer can initiate a BFD session.
- BFD works only for directly-connected neighbors. BFD neighbors must be no more than one IP hop away. Multihop configurations are not supported.
- Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see the [BFD for BGP, page 298](#).

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

Command or Action	Purpose
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>interface type number</code> Example: <pre>Router(config)# interface FastEthernet 6/0</pre>	Enters interface configuration mode.
Step 4 <code>bfd interval milliseconds min_rx milliseconds multiplier interval-multiplier</code> Example: <pre>Router(config-if)# bfd interval 50 min_rx 50 multiplier 5</pre>	Enables BFD on the interface.
Step 5 <code>end</code> Example: <pre>Router(config-if)# end</pre>	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.

- 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
<p>Step 1 <code>enable</code></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.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp tag1</pre>	<p>Specifies a BGP process and enters router configuration mode.</p>
<p>Step 4 <code>neighbor <i>ip-address</i> fall-over bfd</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 172.16.10.2 fall-over bfd</pre>	<p>Enables BFD support for fallover.</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Returns the router to privileged EXEC mode.</p>
<p>Step 6 <code>show bfd neighbors [details]</code></p> <p>Example:</p> <pre>Router# show bfd neighbors detail</pre>	<p>Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.</p>
<p>Step 7 <code>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]]]</code></p> <p>Example:</p> <pre>Router# show ip bgp neighbors</pre>	<p>Displays information about BGP and TCP connections to neighbors.</p>

Monitoring and Troubleshooting BFD for Cisco 7600 Series Routers

To monitor or troubleshoot BFD on Cisco 7600 series routers, 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	<p><code>enable</code></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><code>show bfd neighbors [details]</code></p> <p>Example:</p> <pre>Router# show bfd neighbors details</pre>	<p>(Optional) Displays the BFD adjacency database.</p> <ul style="list-style-type: none"> • The details keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	<p><code>debug bfd [event packet ipc-error ipc-event oir-error oir-event]</code></p> <p>Example:</p> <pre>Router# debug bfd packet</pre>	<p>(Optional) Displays debugging information about BFD packets.</p>

- [What to Do Next, page 331](#)

What to Do Next

For more information about configuring BFD support for another routing protocol see the "Bidirectional Forwarding Detection" configuration guide.

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 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 snmp-server enable traps bgp <code>[[state-changes [all] [backward-trans] [limited]] [threshold prefix]]</code> Example: <pre>Router# snmp-server enable traps bgp</pre>	Enables BGP support for SNMP operations. Entering this command with no keywords or arguments enables support for all BGP events. <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.
Step 4 exit Example: <pre>Router(config)# exit</pre>	Exits global configuration mode, and enters privileged EXEC mode.

Configuring BGP Support for MTR

Before performing the following tasks, you must have configured MTR topologies. For more details, see the "Multi-Topology Routing" feature in Cisco IOS Release 12.2(33)SRB.

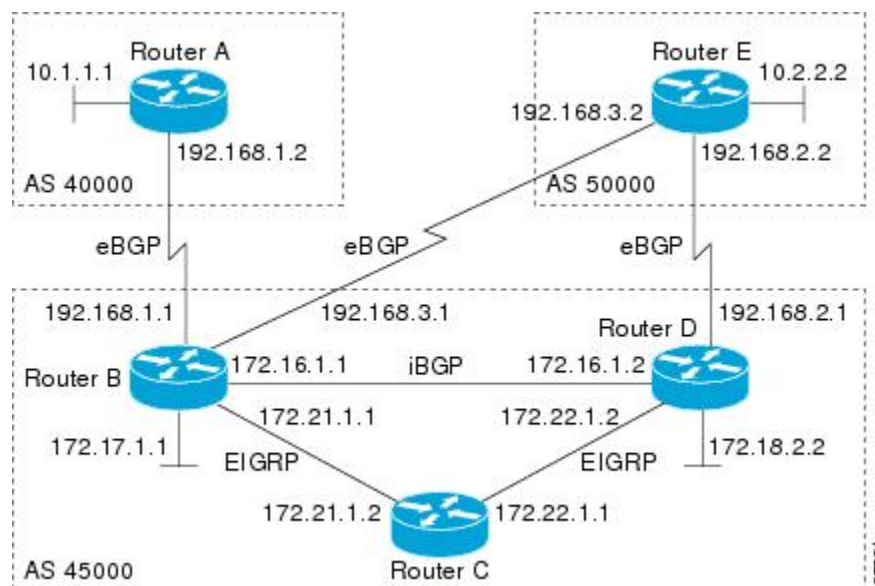
- [Activating an MTR Topology Using BGP, page 333](#)

- [Importing Routes from an MTR Topology Using BGP](#), page 338

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 30 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
>} [<
saf
>]
! 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 [BGP Support for MTR, page 300](#).

- 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> } 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>

Command or Action	Purpose
<p>Step 6 <code>neighbor {ip-address peer-group-name}</code> <code>transport{connection-mode {active passive} </code> <code>path-mtu-discovery multi-session single-</code> <code>session }</code></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.
<p>Step 7 <code>address-family ipv4 [mdt multicast unicast]</code></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.
<p>Step 8 <code>topology {base topology-name}</code></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>
<p>Step 9 <code>bgp tid number</code></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.
<p>Step 10 <code>neighbor ip-address activate</code></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>

Command or Action	Purpose
<p>Step 11 <code>neighbor {ip-address peer-group-name}</code> <code>translate-topology number</code></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.
<p>Step 12 <code>end</code></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>
<p>Step 13 <code>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]]</code></p> <p>Example:</p> <pre>Router# clear ip bgp topology VIDEO 45000</pre>	<p>Resets BGP neighbor sessions under a specified topology or all topologies.</p>
<p>Step 14 <code>show ip bgp topology {* topology} summary</code></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, page 337](#)

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.

- You must be running a Cisco IOS Release 12.2(33)SRB, or later release, on any routers configured for MTR.
- 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	<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>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 TEN permit 10.2.2.0/24</pre>	<p>Configures an IP prefix list.</p> <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	<p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map 10NET</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 named 10NET 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>Router(config-route-map)# match ip address prefix-list TEN</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 prefixes permitted by prefix list TEN.
Step 6	<p>exit</p> <p>Example:</p> <pre>Router(config-route-map)# exit</pre>	<p>Exits route map configuration mode and returns to global configuration mode.</p>

Command or Action	Purpose
<p>Step 7 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
<p>Step 8 <code>scope {global vrf <i>vrf-name</i>}</code></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.
<p>Step 9 <code>address-family ipv4 [mdt multicast unicast]</code></p> <p>Example:</p> <pre>Router(config-router-scope)# address-family ipv4</pre>	<p>Enters router scope address family configuration mode to configure an address family session under BGP.</p> <ul style="list-style-type: none"> • Non-topology-specific configuration parameters are configured in this configuration mode.
<p>Step 10 <code>topology {base <i>topology-name</i>}</code></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>
<p>Step 11 <code>import topology {base <i>topology-name</i>}[route-map <i>map-name</i>]</code></p> <p>Example:</p> <pre>Router(config-router-scope-af-topo)# import topology VOICE route-map 10NET</pre>	<p>(Optional) Configures BGP to move routes from one topology to another on the same router.</p> <ul style="list-style-type: none"> • The route-map keyword can be used to filter routes that moved between topologies.
<p>Step 12 <code>end</code></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>

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the “Connecting to a Service Provider Using External BGP” module.
- If you want to configure some internal BGP features, see the “Configuring Internal BGP Features” chapter of the BGP section of the *Cisco IOS IP Routing Protocols Configuration Guide*.
- If you want to configure BGP neighbor session options, see the “Configuring BGP Neighbor Session Options” module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	“Cisco BGP Overview” module of the <i>Cisco IOS IP Routing Protocols Configuration Guide</i> .
Conceptual and configuration details for basic BGP tasks.	“Configuring a Basic BGP Network” module of the <i>Cisco IOS IP Routing Protocols Configuration Guide</i> .
Information about SNMP and SNMP operations.	“Configuring SNMP Support” section of the <i>Cisco IOS Network Management 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 1657	<i>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIPv2</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 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 4724	<i>Graceful Restart Mechanism for BGP</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 Advanced BGP Features

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

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

Table 19 **Feature Information for Configuring Advanced BGP Features**

Feature Name	Releases	Feature Configuration Information
BGP Graceful Restart per Neighbor	12.2(33)SRC 12.2(33)SB 15.0(1)M 15.0(1)S Cisco IOS XE 3.1.0SG	<p>The BGP Graceful Restart per Neighbor feature enables or disables the BGP graceful restart capability for an individual BGP neighbor, including using peer session templates and BGP peer groups.</p> <p>In Cisco IOS Release 12.2(33)SB, platform support includes the Cisco 10000 series routers.</p> <p>The following commands were introduced or modified by this feature: ha-mode graceful-restart, neighbor ha-mode graceful-restart, show ip bgp neighbors.</p>
BGP MIB Support Enhancements	12.0(26)S 12.2(25)S 12.3(7)T 12.2(33)SRA 12.2(33)SXH	<p>The BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications.</p> <p>The following command was introduced in this feature: snmp-server enable traps bgp.</p>

Feature Name	Releases	Feature Configuration Information
BGP Nonstop Forwarding (NSF) Awareness	12.2(15)T 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified by this feature: bgp graceful-restart, show ip bgp, show ip bgp neighbors.</p>
BGP Selective Address Tracking	12.4(4)T 12.2(33)SRB	<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 Configuration Information
BGP Support for BFD	12.0(31)S 12.4(4)T 12.2(33)SRA 12.2(33)SXH 12.2(33)SB 15.0(1)S	<p data-bbox="1154 289 1507 1050">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 convergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster convergence time.</p> <p data-bbox="1154 1066 1507 1226">The following commands were introduced or modified by this feature: bfd, neighbor fall-over, show bfd neighbors, show ip bgp neighbors.</p>

Feature Name	Releases	Feature Configuration Information
BGP Support for MTR	12.2(33)SRB	<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>

Feature Name	Releases	Feature Configuration Information
BGP Support for Next-Hop Address Tracking	12.0(29)S 12.3(14)T 12.2(33)SXH 15.0(1)S	<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>

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: 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)

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



Configuring Multiprotocol BGP (MP-BGP) Support for CLNS

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

- [Finding Feature Information, page 349](#)
- [Restrictions for Configuring MP-BGP Support for CLNS, page 349](#)
- [Information About Configuring MP-BGP Support for CLNS, page 350](#)
- [How to Configure MP-BGP Support for CLNS, page 353](#)
- [Configuration Examples for MP-BGP Support for CLNS, page 377](#)
- [Additional References, page 386](#)
- [Feature Information for Configuring MP-BGP Support for CLNS, page 387](#)
- [Glossary, page 389](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Restrictions for Configuring MP-BGP Support for CLNS

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

BGP extended communities are not supported by this feature.

The following BGP commands are not supported by this feature:

- **auto-summary**
- **neighbor advertise-map**

- **neighbor distribute-list**
- **neighbor soft-reconfiguration**
- **neighbor unsuppress-map**

Information About Configuring MP-BGP Support for CLNS

- [Address Family Routing Information, page 350](#)
- [Design Features of MP-BGP Support for CLNS, page 350](#)
- [Generic BGP CLNS Network Topology, page 350](#)
- [DCN Network Topology, page 352](#)
- [Benefits of MP-BGP Support for CLNS, page 353](#)

Address Family Routing Information

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

Design Features of MP-BGP Support for CLNS

The configuration of MP-BGP support for CLNS allows BGP to be used as an interdomain routing protocol in networks that use CLNS as the network-layer protocol. This feature was developed to solve a scaling issue with a data communications network (DCN) where large numbers of network elements are managed remotely. For details about the DCN issues and how to implement this feature in a DCN topology, see the [DCN Network Topology, page 352.](#)"

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

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

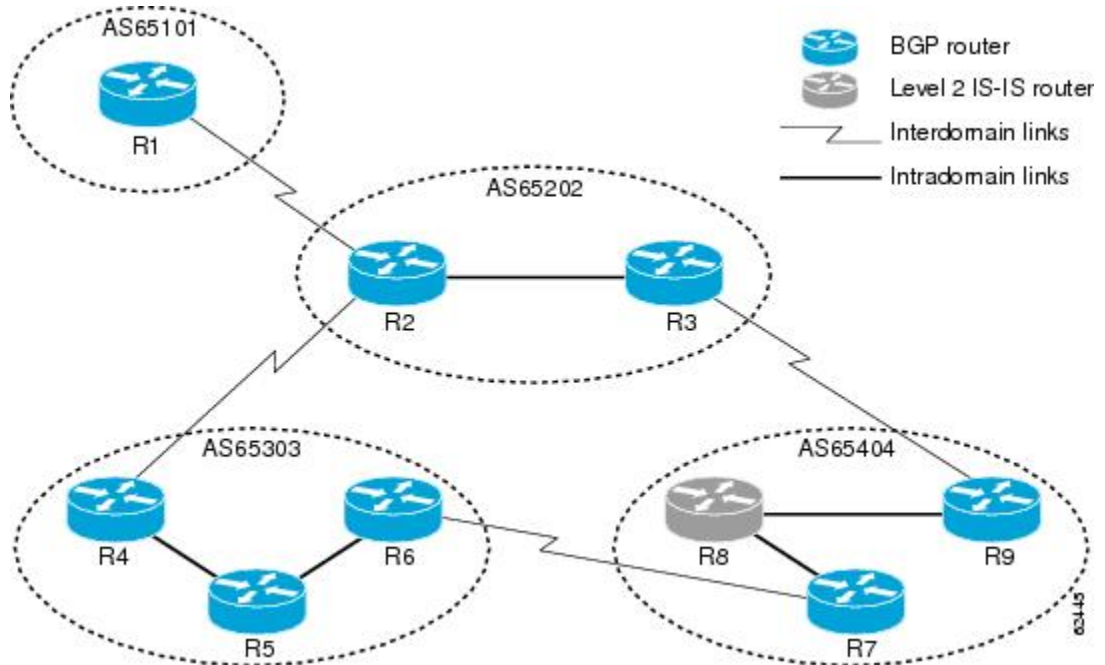
Implementing BGP in routers at the edge of each internal network means that the existing interior protocols need not be changed, minimizing disruption in the network.

Generic BGP CLNS Network Topology

The figure below shows a generic BGP CLNS network containing nine routers that are grouped into four different autonomous systems (in BGP terminology) or routing domains (in OSI terminology). To avoid

confusion, we will use the BGP terminology of autonomous systems because each autonomous system is numbered and therefore more easily identified in the diagram and in the configuration discussion.

Figure 31 *Components in a Generic BGP CLNS Network*



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

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

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

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

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

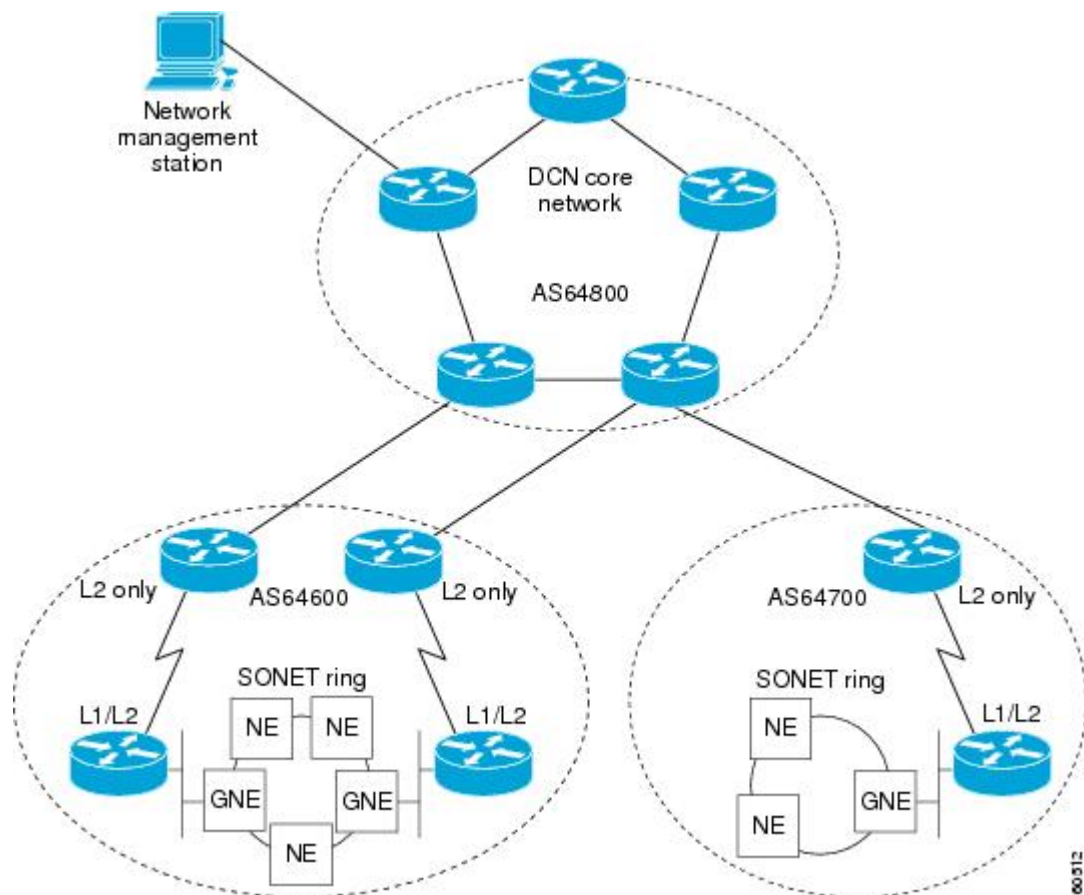
The configuration tasks and examples are based on the generic network design shown in the figure above. Configurations for all the routers in the figure above are listed in .

DCN Network Topology

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

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

Figure 32 Components in a DCN Network



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

Smaller Cisco routers such as the Cisco 2600 series were selected to run as the Level 1-2 routers because shelf space in the central office (CO) of a service provider is very expensive. A Cisco 2600 series router

has limited processing power if it is acting as the Level 1 router for four or five different Level 1 areas. The number of Level 1 areas under this configuration is limited to about 200. The entire Level 2 network is also limited by the speed of the slowest Level 2 router.

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

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

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

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

Benefits of MP-BGP Support for CLNS

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

How to Configure MP-BGP Support for CLNS

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

- [Configuring and Activating a BGP Neighbor to Support CLNS, page 354](#)
- [Configuring an IS-IS Routing Process, page 356](#)
- [Configuring Interfaces That Connect to BGP Neighbors, page 357](#)
- [Configuring Interfaces Connected to the Local OSI Routing Domain, page 359](#)
- [Advertising Networking Prefixes, page 360](#)
- [Redistributing Routes from BGP into IS-IS, page 363](#)
- [Redistributing Routes from IS-IS into BGP, page 364](#)
- [Configuring BGP Peer Groups and Route Reflectors, page 366](#)
- [Filtering Inbound Routes Based on NSAP Prefixes, page 368](#)
- [Filtering Outbound BGP Updates Based on NSAP Prefixes, page 369](#)
- [Originating Default Routes for a Neighboring Routing Domain, page 372](#)
- [Verifying MP-BGP Support for CLNS, page 374](#)
- [Troubleshooting MP-BGP Support for CLNS, page 376](#)

Configuring and Activating a BGP Neighbor to Support CLNS

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

SUMMARY STEPS

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

DETAILED STEPS

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
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>as-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 65101</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 identifies the autonomous system in which the router resides. Valid values are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
<p>Step 4 <code>no bgp default ipv4-unicast</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</p>
<p>Step 5 <code>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i></code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.1.2.2 remote-as 64202</pre>	<p>Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.</p>
<p>Step 6 <code>address-family nsap [<i>unicast</i>]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	<p>Specifies the NSAP address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> The optional unicast keyword specifies the NSAP unicast address prefixes. By default, the router is placed in configuration mode for the unicast NSAP address family if the unicast keyword is not specified with the address-family nsap command.
<p>Step 7 <code>neighbor <i>ip-address</i> activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.1.2.2 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>

Command or Action	Purpose
Step 8 end Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring an IS-IS Routing Process

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

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

SUMMARY STEPS

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

DETAILED STEPS

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.

Command or Action	Purpose
<p>Step 3 <code>router isis area-tag</code></p> <p>Example:</p> <pre>Router(config)# router isis osi-as-101</pre>	<p>Configures an IS-IS routing process and enters router configuration mode for the specified routing process.</p> <ul style="list-style-type: none"> The <i>area-tag</i> argument is a meaningful name for a routing process. It must be unique among all IP and CLNS routing processes for a given router.
<p>Step 4 <code>net network-entity-title</code></p> <p>Example:</p> <pre>Router(config-router)# net 49.0101.1111.1111.1111.1111.00</pre>	<p>Configures a network entity title (NET) for the routing process. If you are configuring multiarea IS-IS, you must specify a NET for each routing process.</p>
<p>Step 5 <code>is-type [level-1 level-1-2 level-2-only]</code></p> <p>Example:</p> <pre>Router(config-router)# is-type level-1</pre>	<p>Configures the router to act as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (interarea) router, or as an interarea router only.</p> <ul style="list-style-type: none"> In multiarea IS-IS configurations, the first instance of the IS-IS routing process configured is by default a Level 1-2 (intra-area and interarea) router. All subsequent IS-IS routing processes on a network running CLNS are configured as Level 1. All subsequent IS-IS routing processes on a network running IP are configured as Level-1-2.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and returns to privileged EXEC mode.</p>

Configuring Interfaces That Connect to BGP Neighbors

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



Note

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

When a neighboring OSI system is found, BGP checks that it is also an eBGP neighbor configured for the NSAP address family. If both the preceding conditions are met, BGP creates a special BGP neighbor route in the CLNS Level 2 prefix routing table. The special BGP neighbor route is automatically redistributed in to the Level 2 routing updates so that all other Level 2 IS-IS routers in the local OSI routing domain know how to reach this eBGP neighbor.

To configure interfaces that are being used to connect with eBGP neighbors, perform the steps in this procedure. These interfaces will normally be directly connected to their eBGP neighbor.

SUMMARY STEPS

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

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 interface <i>type number</i></p> <p>Example:</p> <pre>Router(config)# interface serial 2/0</pre>	<p>Specifies the interface type and number and enters interface configuration mode.</p>
<p>Step 4 ip address <i>ip-address mask</i></p> <p>Example:</p> <pre>Router(config-if)# ip address 10.1.2.2 255.255.255.0</pre>	<p>Configures the interface with an IP address.</p>
<p>Step 5 clns enable</p> <p>Example:</p> <pre>Router(config-if)# clns enable</pre>	<p>Specifies that CLNS packets can be forwarded across this interface. The ES-IS protocol is activated and starts to search for adjacent OSI systems.</p>

Command or Action	Purpose
Step 6 no shutdown Example: Router(config-if)# no shutdown	Turns on the interface.
Step 7 end Example: Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Interfaces Connected to the Local OSI Routing Domain

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

SUMMARY STEPS

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

DETAILED STEPS

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.

Command or Action	Purpose
<p>Step 3 <code>interface type number</code></p> <p>Example:</p> <pre>Router(config)# interface ethernet 0/1</pre>	Specifies the interface type and number and enters interface configuration mode.
<p>Step 4 <code>ip address ip-address mask</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 10.2.3.1 255.255.255.0</pre>	<p>Configures the interface with an IP address.</p> <p>Note This step is required only when the interface needs to communicate with an iBGP neighbor.</p>
<p>Step 5 <code>clns router isis area-tag</code></p> <p>Example:</p> <pre>Router(config-if)# clns router isis osi-as-202</pre>	Specifies that the interface is actively routing IS-IS when the network protocol is ISO CLNS and identifies the area associated with this routing process.
<p>Step 6 <code>ip router isis area-tag</code></p> <p>Example:</p> <pre>Router(config-if)# ip router isis osi-as-202</pre>	<p>Specifies that the interface is actively routing IS-IS when the network protocol is IP and identifies the area associated with this routing process.</p> <p>Note This step is required only when the interface needs to communicate with an iBGP neighbor, and the IGP is IS-IS.</p>
<p>Step 7 <code>no shutdown</code></p> <p>Example:</p> <pre>Router(config-if)# no shutdown</pre>	Turns on the interface.
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Advertising Networking Prefixes

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

SUMMARY STEPS

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

DETAILED STEPS

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>as-number</i> Example: <pre>Router(config)# router bgp 65101</pre>	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4 no bgp default ipv4-unicast Example: <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

Command or Action	Purpose
<p>Step 5 <code>neighbor {ip-address peer-group-name} remote-as as-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.1.2.2 remote-as 64202</pre>	<p>Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.</p>
<p>Step 6 <code>address-family nsap [unicast]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	<p>Specifies the NSAP address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> The optional unicast keyword specifies the NSAP unicast address prefixes. By default, the router is placed in unicast NSAP address family configuration mode if the unicast keyword is not specified with the address-family nsap command.
<p>Step 7 <code>network nsap-prefix [route-map map-tag]</code></p> <p>Example:</p> <pre>Router(config-router-af)# network 49.0101.1111.1111.1111.1111.00</pre>	<p>Advertises a single prefix of the local OSI routing domain and enters it in the BGP routing table.</p> <p>Note It is possible to advertise a single prefix, in which case this prefix could be the unique NSAP address prefix of the local OSI routing domain. Alternatively, multiple longer prefixes, each covering a small portion of the OSI routing domain, can be used to selectively advertise different areas.</p> <ul style="list-style-type: none"> The advertising of NSAP address prefixes can be controlled by using the optional route-map keyword. If no route map is specified, all NSAP address prefixes are redistributed.
<p>Step 8 <code>neighbor ip-address activate</code></p> <p>Example:</p> <pre>Router(config-router-af) neighbor 10.1.2.2 activate</pre>	<p>Specifies that NSAP routing information will be sent to the specified BGP neighbor.</p> <p>Note See the description of the neighbor command in the documents listed in the "Additional References" for more details on the use of this command.</p>
<p>Step 9 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Redistributing Routes from BGP into IS-IS

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

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

SUMMARY STEPS

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

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router isis <i>area-tag</i></p> <p>Example:</p> <pre>Router(config)# router isis osi-as-404</pre>	<p>Configures an IS-IS routing process and enters router configuration mode for the specified routing process.</p> <p>Note You cannot redistribute BGP routes into a Level 1-only IS-IS routing process.</p>
<p>Step 4 net <i>network-entity-title</i></p> <p>Example:</p> <pre>Router(config-router)# net 49.0404.7777.7777.7777.00</pre>	<p>Configures a network entity title (NET) for the routing process. If you are configuring multiarea IS-IS, you must specify a NET for each routing process.</p>

Command or Action	Purpose
<p>Step 5 <code>redistribute protocol as-number [route-type] [route-map map-tag]</code></p> <p>Example:</p> <pre>Router(config-router)# redistribute bgp 65404 clns</pre>	<p>Redistributes NSAP prefix routes from BGP into the CLNS Level 2 routing table associated with the IS-IS routing process when the <i>protocol</i> argument is set to bgp and the <i>route-type</i> argument is set to clns.</p> <ul style="list-style-type: none"> The <i>as-number</i> argument is defined as the autonomous system number of the BGP routing process to be redistributed into CLNS. The redistribution of routes can be controlled by using the optional route-map keyword. If no route map is specified, all BGP routes are redistributed.
<p>Step 6 <code>end</code></p> <p>Example:</p> <pre>Router(config-router)# end</pre>	<p>Exits router configuration mode and returns to privileged EXEC mode.</p>

Redistributing Routes from IS-IS into BGP

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

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

SUMMARY STEPS

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

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></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.

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p>Step 3 <code>router bgp as-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 65202</pre>	Configures a BGP routing process and enters router configuration mode for the specified routing process.
<p>Step 4 <code>no bgp default ipv4-unicast</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
<p>Step 5 <code>address-family nsap [unicast]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	Specifies the NSAP address family and enters address family configuration mode.
<p>Step 6 <code>redistribute protocol [process-id] [route-type] [route-map map-tag]</code></p> <p>Example:</p> <pre>Router(config-router-af)# redistribute isis osi-as-202 clns route-map internal-routes-only</pre>	<p>Redistributes routes from the CLNS Level 2 routing table associated with the IS-IS routing process into BGP as NSAP prefixes when the <i>protocol</i> argument is set to isis and the <i>route-type</i> argument is set to clns.</p> <ul style="list-style-type: none"> The <i>process-id</i> argument is defined as the area name for the relevant IS-IS routing process to be redistributed. The redistribution of routes can be controlled by using the optional route-map keyword. If no route map is specified, all Level 2 routes are redistributed.
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring BGP Peer Groups and Route Reflectors

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

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

SUMMARY STEPS

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

DETAILED STEPS

	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>as-number</i> Example: Router(config)# router bgp 65303	Configures a BGP routing process and enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	<p>no bgp default ipv4-unicast</p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
Step 5	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Router(config-router)# neighbor ibgp-peers peer-group</pre>	Creates a BGP peer group.
Step 6	<p>neighbor <i>peer-group-name</i> remote-as <i>as-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor ibgp-peers remote-as 65303</pre>	Adds the peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.
Step 7	<p>address-family nsap [unicast]</p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	Specifies the NSAP address family and enters address family configuration mode.
Step 8	<p>neighbor <i>peer-group-name</i> route-reflector-client</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor ibgp-peers route-reflector-client</pre>	Configures the router as a BGP route reflector and configures the specified peer group as its client.
Step 9	<p>neighbor <i>ip-address</i> peer-group <i>peer-group</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.4.5.4 peer-group ibgp-peers</pre>	Assigns a BGP neighbor to a BGP peer group.

Command or Action	Purpose
Step 10 <code>end</code> Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Filtering Inbound Routes Based on NSAP Prefixes

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

Command or Action	Purpose
<p>Step 3 <code>router bgp as-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 65200</pre>	Configures a BGP routing process and enters router configuration mode for the specified routing process.
<p>Step 4 <code>no bgp default ipv4-unicast</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
<p>Step 5 <code>address-family nsap [unicast]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	Specifies the address family and enters address family configuration mode.
<p>Step 6 <code>neighbor {ip-address peer-group-name} prefix-list {clns-filter-expr-name clns-filter-set-name} in</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.23.4.1 prefix-list abc in</pre>	<p>Specifies a CLNS filter set or CLNS filter expression to be used to filter inbound BGP routes.</p> <ul style="list-style-type: none"> The <code>clns-filter-expr-name</code> argument is defined with the clns filter-expr configuration command. The <code>clns-filter-set-name</code> argument is defined with the clns filter-set configuration command.
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Filtering Outbound BGP Updates Based on NSAP Prefixes

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

SUMMARY STEPS

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

DETAILED STEPS

	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	clns filter-set <i>name</i> [deny] template Example: <pre>Router(config)# clns filter-set routes0404 deny 49.0404...</pre>	Defines a NSAP prefix match for a deny condition for use in CLNS filter expressions. <ul style="list-style-type: none"> • In this example, a deny action is returned if an address starts with 49.0404.

	Command or Action	Purpose
Step 4	<p>clns filter-set <i>name</i> [permit] template</p> <p>Example:</p> <pre>Router(config)# clns filter-set routes0404 permit 49...</pre>	<p>Defines a NSAP prefix match for a permit condition for use in CLNS filter expressions.</p> <ul style="list-style-type: none"> In this example, a permit action is returned if an address starts with 49. <p>Note Although the permit example in this step allows all NSAP addresses starting with 49, the match condition in Step 3 is processed first so the NSAP addresses starting with 49.0404 are still denied.</p>
Step 5	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 65404</pre>	<p>Configures a BGP routing process and enters router configuration mode for the specified routing process.</p>
Step 6	<p>no bgp default ipv4-unicast</p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.</p>
Step 7	<p>neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Router(config-router)# neighbor ebgp- peers peer-group</pre>	<p>Creates a BGP peer group.</p> <ul style="list-style-type: none"> In this example, the BGP peer group named ebgp-peers is created.
Step 8	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor ebgp- peers remote-as 65303</pre>	<p>Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.</p> <ul style="list-style-type: none"> In this example, the peer group named ebgp-peers is added to the BGP neighbor table.
Step 9	<p>address-family nsap [unicast]</p> <p>Example:</p> <pre>Router(config-router)# address-family nsap</pre>	<p>Specifies the NSAP address family and enters address family configuration mode.</p>

Command or Action	Purpose
<p>Step 10 neighbor {<i>ip-address</i> <i>peer-group-name</i>} prefix-list {<i>clns-filter-expr-name</i> <i>clns-filter-set-name</i>} out</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor ebgp-peers prefix-list routes0404 out</pre>	<p>Specifies a CLNS filter set or CLNS filter expression to be used to filter outbound BGP updates.</p> <ul style="list-style-type: none"> The <i>clns-filter-expr-name</i> argument is defined with the clns filter-expr configuration command. The <i>clns-filter-set-name</i> argument is defined with the clns filter-set configuration command. In this example, the filter set named routes0404 was created in Step 3 and Step 4.
<p>Step 11 neighbor <i>ip-address</i> peer-group <i>peer-group</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.6.7.8 peer-group ebgp-peers</pre>	<p>Assigns a BGP neighbor to a BGP peer group.</p>
<p>Step 12 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>

Originating Default Routes for a Neighboring Routing Domain

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

SUMMARY STEPS

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

DETAILED STEPS

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

Command or Action	Purpose
Step 7 <code>end</code> Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying MP-BGP Support for CLNS

To verify the configuration, use the **show running-config** EXEC command. Sample output is located in the [Implementing MP-BGP Support for CLNS Example, page 381](#). To verify that the Multiprotocol BGP (MP-BGP) Support for CLNS feature is working, perform the following steps.

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **show clns neighbors**

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

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

Example:

```
Router# show clns neighbors
Tag osi-as-202:
System Id      Interface  SNPA          State  Holdtime  Type  Protocol
r1             Se2/0     *HDLC*        Up     274       IS   ES-IS
```

```

r3          Et0/1          0002.16de.8481      Up      9          L2    IS-IS
r4          Se2/2          *HDLC*              Up     275        IS    ES-IS

```

Step 2 **show clns route**

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

Example:

```

Router# show clns route
Codes: C - connected, S - static, d - DecnetIV
       I - ISO-IGRP, i - IS-IS, e - ES-IS
       B - BGP, b - eBGP-neighbor
C 49.0202.2222 [2/0], Local IS-IS Area
C 49.0202.2222.2222.2222.00 [1/0], Local IS-IS NET
b 49.0101.1111.1111.1111.00 [15/10]
   via r1, Serial2/0
i 49.0202.3333 [110/10]
   via r3, Ethernet0/1
b 49.0303.4444.4444.4444.00 [15/10]
   via r4, Serial2/2
B 49.0101 [20/1]
   via r1, Serial2/0
B 49.0303 [20/1]
   via r4, Serial2/2
B 49.0404 [200/1]
   via r9
i 49.0404.9999.9999.9999.00 [110/10]
   via r3, Ethernet0/1

```

Step 3 **show bgp nsap unicast summary**

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

Example:

```

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

```

Step 4 Enter the **show bgp nsap unicast** command to display all the NSAP prefix routes that the local router has discovered. In the following example of output from router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), a single valid route to prefix 49.0101 is shown. Two valid routes--marked by a *--are shown for the prefix 49.0404. The second route is marked with a *>i sequence, representing the best route to this prefix.

Example:

```

Router# show bgp nsap unicast
BGP table version is 3, local router ID is 192.168.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

```

```

      r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          Next Hop           Metric LocPrf Weight Path
*> 49.0101         49.0101.1111.1111.1111.1111.00
                                     0 65101 i
* i49.0202.2222    49.0202.3333.3333.3333.3333.00
                                     100   0 ?
*>                49.0202.2222.2222.2222.2222.00
                                     32768 ?
* i49.0202.3333    49.0202.3333.3333.3333.3333.00
                                     100   0 ?
*>                49.0202.2222.2222.2222.2222.00
                                     32768 ?
*> 49.0303         49.0303.4444.4444.4444.4444.00
                                     0 65303 i
* 49.0404          49.0303.4444.4444.4444.4444.00
                                     0 65303 65404 i
*>i              49.0404.9999.9999.9999.9999.00
                                     100   0 65404 i

```

Troubleshooting MP-BGP Support for CLNS

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

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

SUMMARY STEPS

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

DETAILED STEPS

- Step 1** Attach a console directly to a router running the Cisco IOS software release that includes the Multiprotocol BGP (MP-BGP) Support for CLNS feature.

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

- Step 2** **no logging console**
This command disables all logging to the console terminal.
- Step 3** Use Telnet to access a router port.
- Step 4** **enable**
Enter this command to access privileged EXEC mode.
- Step 5** **terminal monitor**
This command enables logging on the virtual terminal.
- Step 6** **debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]
Enter only specific **debug bgp nsap unicast** commands to isolate the output to a certain subcomponent and minimize the load on the processor. Use appropriate arguments and keywords to generate more detailed debug information on specified subcomponents.
- Step 7** **no terminal monitor**
This command disables logging on the virtual terminal.
- Step 8** **no debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]
Enter the specific **no debug bgp nsap unicast** command when you are finished.
- Step 9** **logging console**
This command reenables logging to the console.
-

Configuration Examples for MP-BGP Support for CLNS

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

- [Configuring and Activating a BGP Neighbor to Support CLNS Example, page 378](#)
- [Configuring an IS-IS Routing Process Example, page 378](#)
- [Configuring Interfaces Example, page 378](#)
- [Advertising Networking Prefixes Example, page 378](#)
- [Redistributing Routes from BGP into IS-IS Example, page 379](#)
- [Redistributing Routes from IS-IS into BGP Example, page 379](#)
- [Configuring BGP Peer Groups and Route Reflectors Example, page 379](#)
- [Filtering Inbound Routes Based on NSAP Prefixes Example, page 380](#)
- [Filtering Outbound BGP Updates Based on NSAP Prefixes Example, page 380](#)
- [Originating a Default Route and Outbound Route Filtering Example, page 380](#)
- [Implementing MP-BGP Support for CLNS Example, page 381](#)

Configuring and Activating a BGP Neighbor to Support CLNS Example

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

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

Configuring an IS-IS Routing Process Example

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

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

The default IS-IS routing process level is used.

Configuring Interfaces Example

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

```
interface serial 2/0
 ip address 10.1.2.2 255.255.255.0
 clns enable
 no shutdown
!
interface ethernet 0/1
 ip address 10.2.3.1 255.255.255.0
 clns router isis osi-as-202
 no shutdown
```

Advertising Networking Prefixes Example

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

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



```
network 49.0101...
neighbor 10.1.2.2 activate
```

Redistributing Routes from BGP into IS-IS Example

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

Router R7

```
router isis osi-as-404
 net 49.0404.7777.7777.7777.00
 redistribute bgp 65404 clns
```

Router R9

```
router isis osi-as-404
 net 49.0404.9999.9999.9999.00
 redistribute bgp 65404 clns
```

Redistributing Routes from IS-IS into BGP Example

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

```
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
 match clns address internal-routes
!
router isis osi-as-202
 net 49.0202.2222.2222.2222.00
!
router bgp 65202
 no bgp default ipv4-unicast
 address-family nsap
 redistribute isis osi-as-202 clns route-map internal-routes-only
```

Configuring BGP Peer Groups and Route Reflectors Example

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

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

```
router bgp 65303
no bgp default ipv4-unicast
neighbor ibgp-peers peer-group
neighbor ibgp-peers remote-as 65303
address-family nsap
neighbor ibgp-peers route-reflector-client
neighbor 10.4.5.4 peer-group ibgp-peers
neighbor 10.5.6.6 peer-group ibgp-peers
exit-address-family
```

Filtering Inbound Routes Based on NSAP Prefixes Example

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

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

Filtering Outbound BGP Updates Based on NSAP Prefixes Example

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

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

Originating a Default Route and Outbound Route Filtering Example

In the figure below, autonomous system AS65101 is connected to only one other autonomous system, AS65202. Router R2 in AS65202 provides the connectivity to the rest of the network for autonomous system AS65101 by sending a default route to R1. Any packets from Level 1 routers within autonomous system AS65101 with destination NSAP addresses outside the local Level 1 network are sent to R1, the nearest Level 2 router. Router R1 forwards the packets to router R2 using the default route.

In the following example, the router R2, shown in the figure below, in the autonomous system AS65202 is configured to generate a default route for router R1 in the autonomous system AS65101, and an outbound

filter is created to send only the default route NSAP addressing information in the BGP update messages to router R1.

```

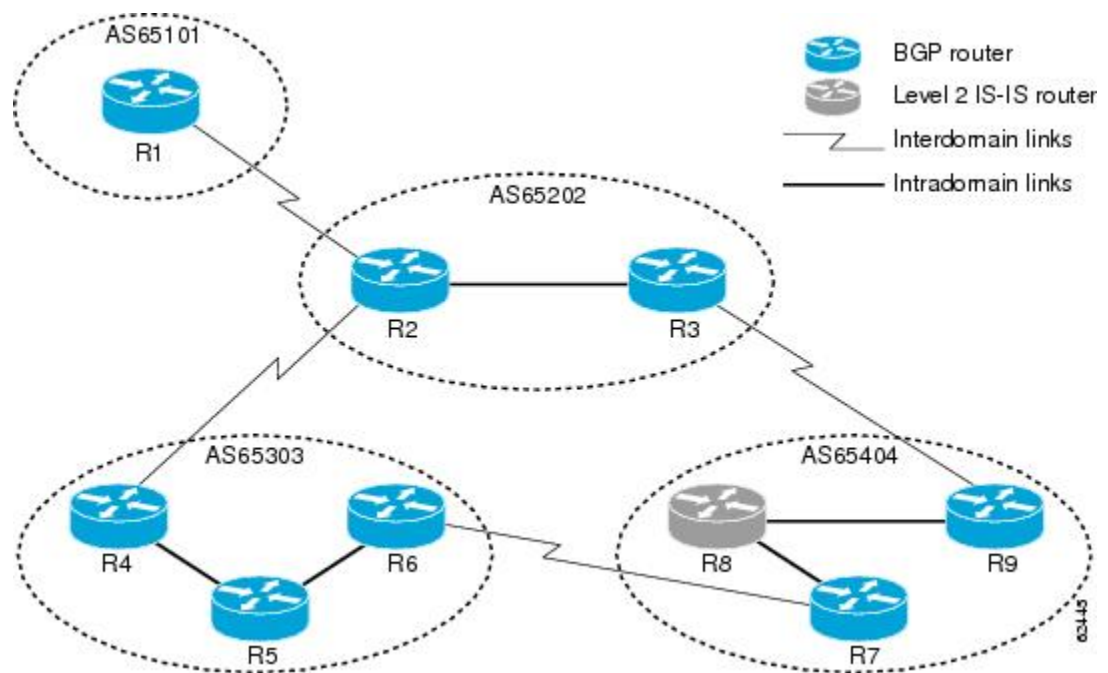
clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
router bgp 65202
no bgp default ipv4-unicast
neighbor 10.1.2.1 remote-as 64101
address-family nsap
network 49.0202...
neighbor 10.1.2.1 activate
neighbor 10.1.2.1 default-originate
neighbor 10.1.2.1 prefix-list default-prefix-only out

```

Implementing MP-BGP Support for CLNS Example

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

Figure 33 Components in a Generic BGP CLNS Network



If you need more details about commands used in the following examples, see the configuration tasks earlier in this document and the documents listed in the [Additional References](#), page 386.

- [Autonomous System AS65101](#), page 382
- [Autonomous System AS65202](#), page 382
- [Autonomous System AS65303](#), page 383
- [Autonomous System AS65404](#), page 384

Autonomous System AS65101

Router 1

```

clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
router isis osi-as-101
 net 49.0101.1111.1111.1111.1111.00
!
router bgp 65101
 no bgp default ipv4-unicast
 neighbor 10.1.2.2 remote-as 65202
 address-family nsap
  neighbor 10.1.2.2 activate
  neighbor 10.1.2.2 prefix-list default-prefix-only in
 network 49.0101...
 exit-address-family
!
interface serial 2/0
 ip address 10.1.2.1 255.255.255.0
 clns enable
 no shutdown

```

Autonomous System AS65202

Router 2

```

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

```

```
clns enable
no shutdown
```

Router 3

```
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
  match clns address internal-routes
!
router isis osi-as-202
  net 49.0202.3333.3333.3333.3333.00
!
router bgp 65202
  no bgp default ipv4-unicast
  neighbor 10.2.3.2 remote-as 65202
  neighbor 10.3.9.9 remote-as 65404
  address-family nsap
    neighbor 10.2.3.2 activate
    neighbor 10.3.9.9 activate
  redistribute isis osi-as-202 clns route-map internal-routes-only
  exit-address-family
!
interface ethernet 0/1
  ip address 10.2.3.3 255.255.255.0
  clns router isis osi-as-202
  no shutdown
!
interface serial 2/2
  ip address 10.3.9.3 255.255.255.0
  clns enable
  no shutdown
```

Autonomous System AS65303

Router 4

```
router isis osi-as-303
  net 49.0303.4444.4444.4444.4444.00
!
router bgp 65303
  no bgp default ipv4-unicast
  neighbor 10.2.4.2 remote-as 65202
  neighbor 10.4.5.5 remote-as 65303
  address-family nsap
    no synchronization
    neighbor 10.2.4.2 activate
    neighbor 10.4.5.5 activate
  network 49.0303...
  exit-address-family
!
interface ethernet 0/2
  ip address 10.4.5.4 255.255.255.0
  clns router isis osi-as-303
  no shutdown
!
interface serial 2/3
  ip address 10.2.4.4 255.255.255.0
  clns enable
  no shutdown
```

Router 5

```
router isis osi-as-303
  net 49.0303.5555.5555.5555.5555.00
!
```

```

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

```

Router 6

```

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

```

Autonomous System AS65404

Router 7

```

clns filter-set external-routes deny 49.0404...
clns filter-set external-routes permit 49...
!
route-map noexport permit 10
  match clns address external-routes
  set community noexport
!
router isis osi-as-404
  net 49.0404.7777.7777.7777.00
  redistribute bgp 404 clns
!
router bgp 65404
  no bgp default ipv4-unicast
  neighbor 10.6.7.6 remote-as 65303
  neighbor 10.8.9.9 remote-as 65404
  address-family nsap
    neighbor 10.6.7.6 activate
    neighbor 10.8.9.9 activate

```

```

neighbor 10.8.9.9 send-community
neighbor 10.8.9.9 route-map noexport out
network 49.0404...
!
interface ethernet 1/0
ip address 10.7.8.7 255.255.255.0
clns router isis osi-as-404
ip router isis osi-as-404
no shutdown
!
interface serial 2/3
ip address 10.6.7.7 255.255.255.0
clns enable
no shutdown

```

Router 8

```

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

```

Router 9

```

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

```

Additional References

The following sections provide references related to the Multiprotocol BGP (MP-BGP) Support for CLNS feature.

Related Documents

Related Topic	Document Title
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
CLNS commands	<i>Cisco IOS ISO CLNS Command Reference</i>

Standards

Standard	Title
ISO/IEC 8473	<i>ISO CLNP: Connectionless Network Protocol (ISO-IP)</i> . Protocol for providing the connectionless-mode network service.
ISO/IEC 9542	<i>End System to Intermediate System Protocol (ESIS)</i> . End system to Intermediate system routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473).
ISO/IEC 10589	<i>IS-IS, Intermediate System-to-Intermediate System</i> . Intermediate system to Intermediate system intradomain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473).

MIBs

MIB	MIBs Link
None.	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 1700	<i>Assigned Numbers</i>
RFC 1771	<i>A Border Gateway Protocol 4 (BGP-4)</i>

RFC	Title
RFC 1997	<i>BGP Communities Attribute</i>
RFC 2042	<i>Registering New BGP Attribute Types</i>
RFC 2439	<i>BGP Route Flap Dampening</i>
RFC 2842	<i>Capabilities Advertisement with BGP-4</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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for Configuring MP-BGP Support for CLNS

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

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

Table 20 Feature Information for MP-BGP Support for CLNS

Feature Name	Releases	Feature Information
Multiprotocol BGP (MP-BGP) Support for CLNS	12.2(8)T 12.2(33)SRB	<p>The Multiprotocol BGP (MP-BGP) Support for CLNS feature provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks.</p> <p>In Release 12.2(8)T, this feature was introduced on the following platforms:</p> <ul style="list-style-type: none"> • Cisco 2600 series • Cisco 3600 series • Cisco 7100 series • Cisco 7200 series • Cisco 7500 series • Cisco uBR7200 series <p>In Release 12.2(33)SRB, this feature was introduced on the Cisco 7600 Series.</p> <p>The following commands were introduced or modified by this feature: address-family nsap, clear bgp nsap, clear bgp nsap dampening, clear bgp nsap external, clear bgp nsap flap-statistics, clear bgp nsap peer-group, debug bgp nsap, debug bgp nsap dampening, debug bgp nsap updates, neighbor prefix-list, network (BGP and multiprotocol BGP), redistribute (BGP to ISO ISIS), redistribute (ISO ISIS to BGP), show bgp nsap, show bgp nsap community, show bgp nsap community-list, show bgp nsap dampened-paths, show bgp nsap filter-list, show bgp nsap</p>

Feature Name	Releases	Feature Information
		flap-statistics, show bgp nsap inconsistent-as, show bgp nsap neighbors, show bgp nsap paths, show bgp nsap quote-regexp, show bgp nsap regexp, show bgp nsap summary.

Glossary

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

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

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

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

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

DCC --data communications channel.

DCN --data communications network.

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

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

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

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

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

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

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

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

OSI --Open System Interconnection. International standardization program created by ISO and ITU-T to develop standards for data networking that facilitate multivendor equipment interoperability.

routing domain --The OSI term that is equivalent to autonomous system for BGP.

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

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

Cisco and the Cisco logo are trademarks 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: 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)

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



BGP 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 391](#)
- [Prerequisites for BGP Link Bandwidth, page 391](#)
- [Restrictions for BGP Link Bandwidth, page 392](#)
- [Information About BGP Link Bandwidth, page 392](#)
- [How to Configure BGP Link Bandwidth, page 393](#)
- [Configuration Examples for BGP Link Bandwidth, page 395](#)
- [Where to Go Next, page 399](#)
- [Additional References, page 399](#)
- [Feature Information for BGP Link Bandwidth, page 400](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP Link Bandwidth

- BGP load balancing or multipath load balancing must be configured before BGP Link Bandwidth feature is enabled.
- BGP extended community exchange must be enabled between iBGP neighbors to which the link bandwidth attribute is to be advertised.
- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Restrictions for BGP Link Bandwidth

- The BGP Link Bandwidth feature can be configured only under IPv4 and VPNv4 address family sessions.
- BGP can originate the link bandwidth community only for directly connected links to eBGP neighbors.
- Both iBGP and eBGP load balancing are supported in IPv4 and VPNv4 address families. However, eiBGP load balancing is supported only in VPNv4 address families.

Information About BGP Link Bandwidth

- [BGP Link Bandwidth Overview](#), page 392
- [Link Bandwidth Extended Community Attribute](#), page 392
- [Benefits of the BGP Link Bandwidth Feature](#), page 392

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, page 393](#)

Configuring and Verifying BGP Link Bandwidth

To configure the BGP Link Bandwidth feature, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4**
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: Router> 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: 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 to create or configure a BGP routing process.

Command or Action	Purpose
<p>Step 4 <code>address-family ipv4</code></p> <p>Example:</p> <pre>Router(config-router-af)# address-family ipv4</pre>	<p>Enters address family configuration mode.</p>
<p>Step 5 <code>address-family ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4</pre>	<p>The BGP Link Bandwidth feature is supported only under the IPv4 and VPNv4 address families.</p>
<p>Step 6 <code>bgp dmzlink-bw</code></p> <p>Example:</p> <pre>Router(config-router-af)# bgp dmzlink-bw</pre>	<p>Configures BGP to distribute traffic proportionally to the bandwidth of the link.</p> <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.
<p>Step 7 <code>neighbor ip-address dmzlink-bw</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 172.16.1.1 dmzlink-bw</pre>	<p>Configures BGP to include the link bandwidth attribute for routes learned from the external interface specified IP address.</p> <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.
<p>Step 8 <code>neighbor ip-address send-community [both extended standard]</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.10.10.1 send-community extended</pre>	<p>(Optional) Enables community or extended community exchange with the specified neighbor.</p> <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.
<p>Step 9 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode, and enters privileged EXEC mode.</p>

Command or Action	Purpose
<p>Step 10 <code>show ip bgp ip-address [longer-prefixes [injected] shorter-prefixes [mask-length]]</code></p> <p>Example:</p> <pre>Router# show ip bgp 10.0.0.0</pre>	<p>(Optional) Displays information about the TCP and BGP connections to neighbors.</p> <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.
<p>Step 11 <code>show ip route ip-address [mask] [longer-prefixes] protocol [process-id] [list access-list-number access-list-name] static download</code></p> <p>Example:</p> <pre>Router# 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, page 395](#)
- [Example: Verifying BGP Link Bandwidth, page 398](#)

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.



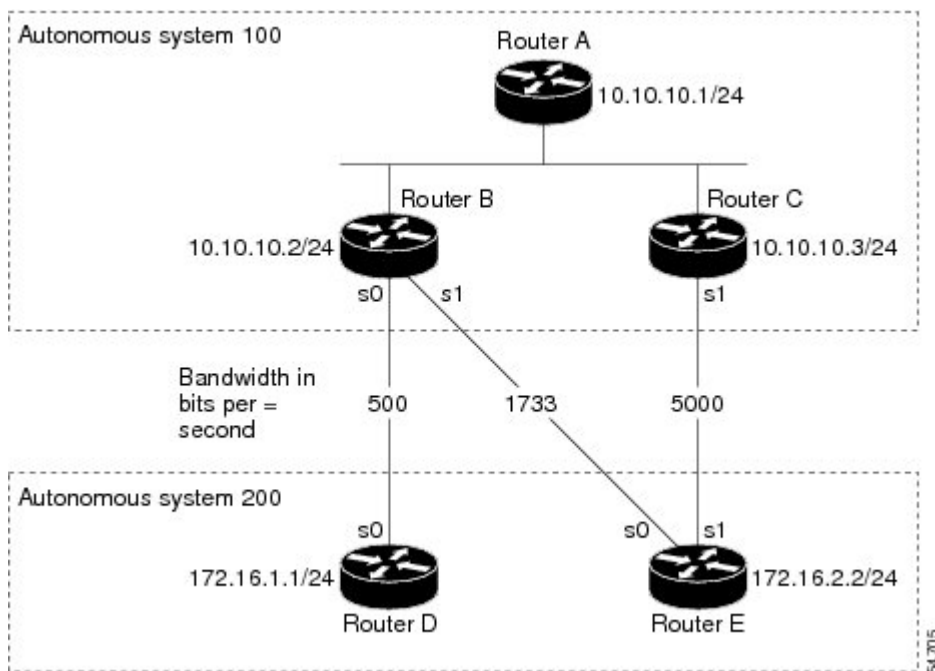
Note

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 34 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
  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, refer to the following document: "BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN".

For more information about the iBGP Multipath Load Sharing feature, refer to the following document: "iBGP Multipath Load Sharing".

Additional References

The following sections provide references related to the BGP Link Bandwidth feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
CEF configuration tasks	"Cisco Express Forwarding Overview" module

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 21 **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>

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: 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)

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



iBGP Multipath Load Sharing

This feature module describes the iBGP Multipath Load Sharing feature.

- [Finding Feature Information, page 403](#)
- [Restrictions for iBGP Multipath Load Sharing, page 403](#)
- [Information about iBGP Multipath Load Sharing, page 404](#)
- [How To Configure iBGP Multipath Load Sharing, page 405](#)
- [Configuration Examples for iBGP Multipath Load Sharing, page 408](#)
- [Additional References, page 410](#)
- [Command Reference, page 411](#)
- [Feature Information for iBGP Multipath Load Sharing, page 411](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

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.
- The iBGP Multipath Load Sharing feature is supported for the following platforms in Cisco IOS Release 12.2(14)S:
 - Cisco 7200 series
 - Cisco 7400 series
 - Cisco 7500 series

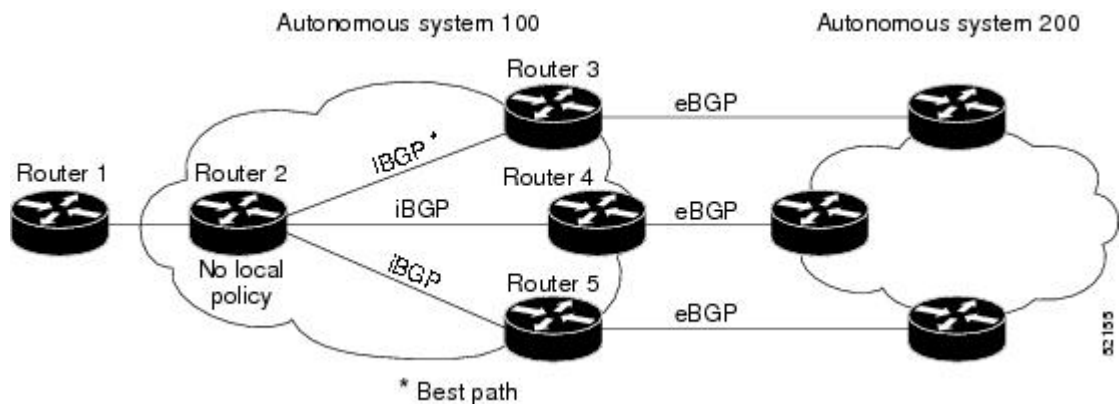
Information about iBGP Multipath Load Sharing

- [iBGP Multipath Load Sharing Overview, page 404](#)
- [Benefits of iBGP Multipath Load Sharing, page 405](#)

iBGP Multipath Load Sharing Overview

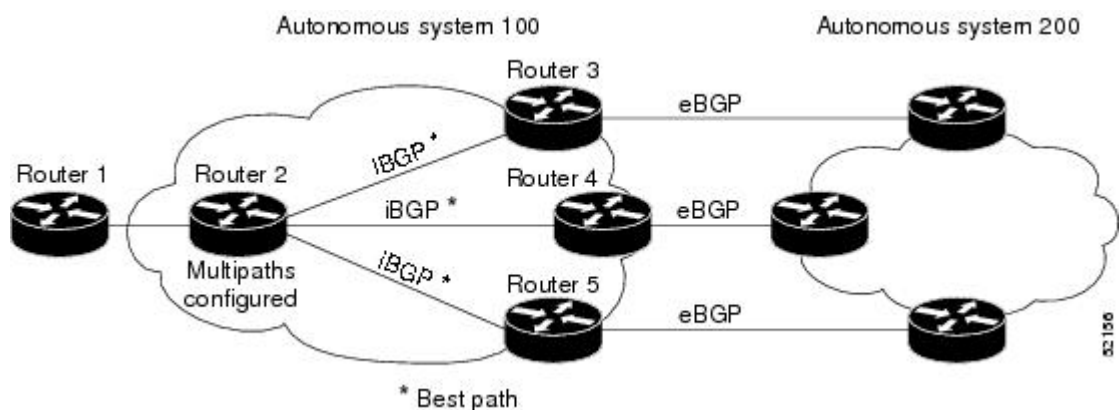
When a Border Gateway Protocol (BGP) speaking router with no local policy configured receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router will choose one iBGP path as the best path. The best path is then installed in the IP routing table of the router. For example, in the figure below, although there are three paths to autonomous system 200, Router 2 determines that one of the paths to autonomous system 200 is the best path and uses this path only to reach autonomous system 200.

Figure 35 Non-MPLS Topology with One Best Path



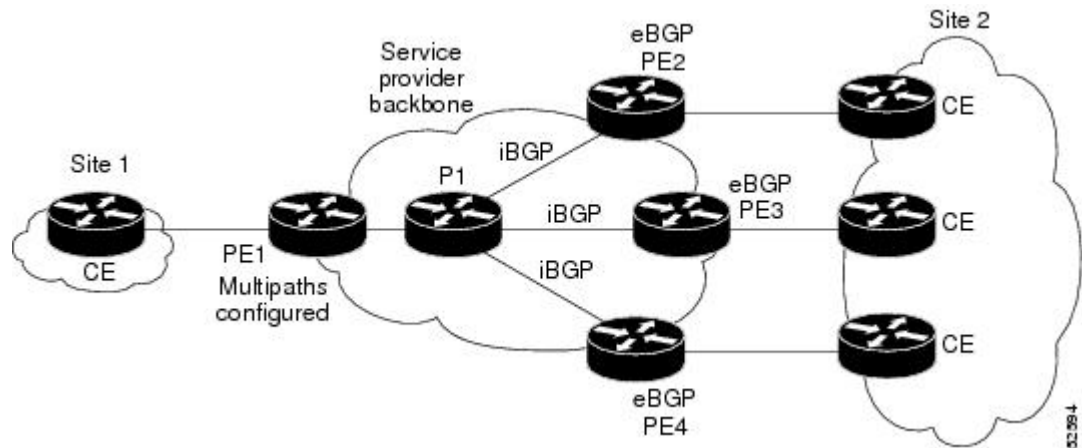
The iBGP Multipath Load Sharing feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router. For example, on router 2 in the figure below, the paths to routers 3, 4, and 5 are configured as multipaths and can be used to reach autonomous system 200, thereby equally sharing the load to autonomous system 200.

Figure 36 Non-MPLS Topology with Three Multipaths



The iBGP Multipath Load Sharing feature functions similarly in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) with a service provider backbone. For example, on router PE1 in the figure below, the paths to routers PE2, PE3, and PE4 can be selected as multipaths and can be used to equally share the load to site 2.

Figure 37 *MPLS VPN with Three Multipaths*



For multiple paths to the same destination to be considered as multipaths, the following criteria must be met:

- All attributes must be the same. The attributes include weight, local preference, autonomous system path (entire attribute and not just length), origin code, Multi Exit Discriminator (MED), and Interior Gateway Protocol (IGP) distance.
- The next hop router for each multipath must be different.

Even if the criteria are met and multiple paths are considered multipaths, the BGP speaking router will still designate one of the multipaths as the best path and advertise this best path to its neighbors.

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, page 405](#)
- [Verifying iBGP Multipath Load Sharing, page 406](#)
- [Monitoring and Maintaining iBGP Multipath Load Sharing, page 408](#)

Configuring iBGP Multipath Load Sharing

To configure the iBGP Multipath Load Sharing feature, use the following command in router configuration mode:

Command	Purpose
Router(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:

```
Router# show ip bgp 10.22.22.0
BGP routing table entry for 10.22.22.0/24, version 119
Paths:(6 available, best #1)
Multipath:iBGP
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
Router# 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:

```

Router# show ip route 10.22.22.0
Routing entry for 10.22.22.0/24
  Known via "bgp 1", distance 200, metric 0
  Tag 22, type internal
  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
Router# 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 vpnv4 all *ip-prefix*** EXEC command are included in the routing information. (The routing information is displayed after performing Step 3.)

Monitoring and Maintaining iBGP Multipath Load Sharing

To display iBGP Multipath Load Sharing information, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show ip bgp <i>ip-prefix</i>	Displays attributes and multipaths for a network in a non-MPLS topology.
Router# show ip bgp vpnv4 all <i>ip-prefix</i>	Displays attributes and multipaths for a network in an MPLS VPN.
Router# show ip route <i>ip-prefix</i>	Displays routing information for a network in a non-MPLS topology.
Router# show ip route vrf <i>vrf-name ip-prefix</i>	Displays routing information for a network in an MPLS VPN.

Configuration Examples for iBGP Multipath Load Sharing

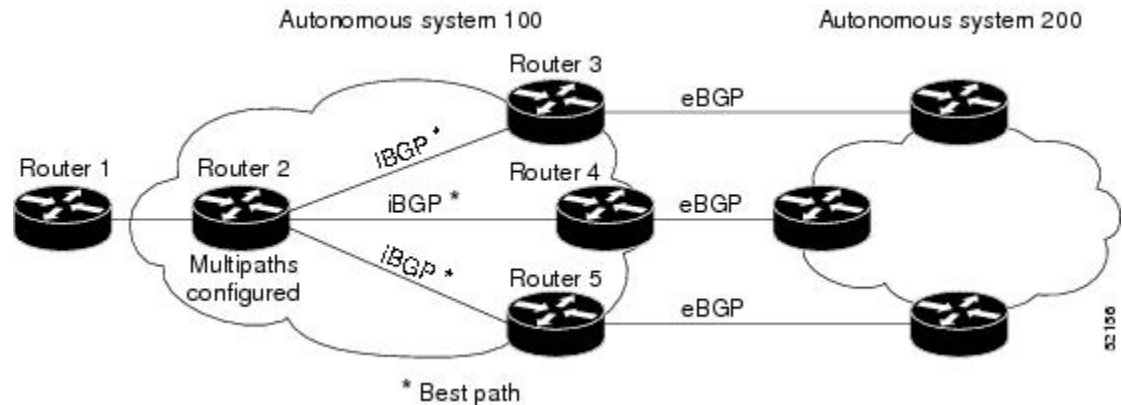
The examples assume that the appropriate attributes for each path are equal and that the next hop router for each multipath is different.

- [Example iBGP Multipath Load Sharing in a Non-MPLS Topology, page 409](#)
- [Example iBGP Multipath Load Sharing in an MPLS VPN Topology, page 409](#)

Example iBGP Multipath Load Sharing in a Non-MPLS Topology

The following example shows how to set up the iBGP Multipath Load Sharing feature in a non-MPLS topology (see the figure below).

Figure 38 *Non-MPLS Topology Example*



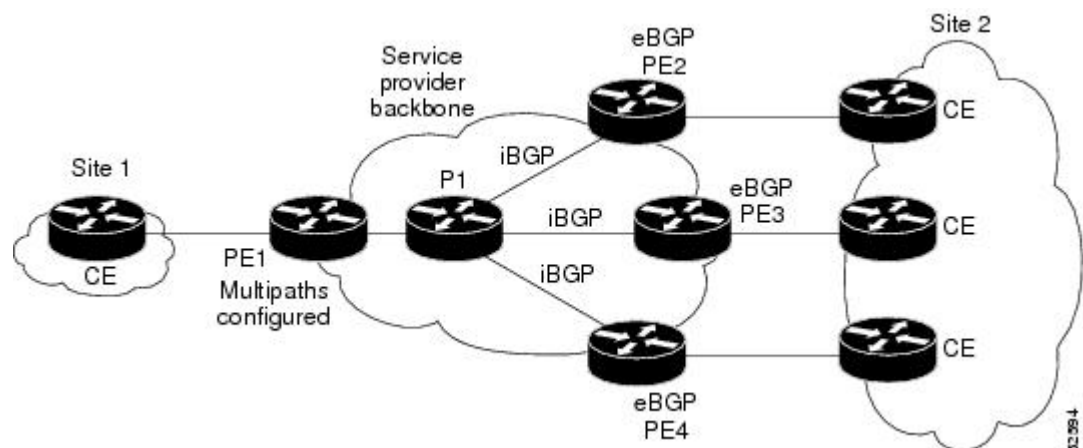
Router 2 Configuration

```
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 39 *MPLS VPN Topology Example*



Router PE1 Configuration

```
router bgp 100
```

```
address-family ipv4 unicast vrf site2
maximum-paths ibgp 3
```

Additional References

The following sections provide references related to the iBGP Multipath Load Sharing feature.

Related Documents

Related Topic	Document Title
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"
Advertising the bandwidth of an autonomous system exit link as an extended community	" BGP Link Bandwidth"
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

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
<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/techsupport</p>

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Routing: BGP Command Reference*. For information about all Cisco IOS commands, go to the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or to the *Cisco IOS Master Commands List*.

New Commands

- **maximum-paths ibgp**

Modified Commands

- **show ip bgp**
- **show ip bgp vpnv4**
- **show ip route**
- **show ip route vrf**

Feature Information for iBGP Multipath Load Sharing

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

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

Table 22 **Feature Information for iBGP Multipath Load Sharing**

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 commands were introduced or modified: maximum-paths ibgp, show ip bgp, show ip bgp vpv4, show ip route, show ip route vrf.</p>

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: 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)

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



BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for 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 413](#)
- [Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 414](#)
- [Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 414](#)
- [Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 414](#)
- [How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 416](#)
- [Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature, page 419](#)
- [Where to Go Next, page 420](#)
- [Additional References, page 420](#)
- [Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, page 422](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Load Balancing is Configured Under CEF

Cisco Express Forwarding (CEF) or distributed CEF (dCEF) must be enabled on all participating routers.

Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Address Family Support

This feature is configured on a per VPN routing and forwarding instance (VRF) basis. This feature can be configured under only the IPv4 VRF address family.

Memory Consumption Restriction

Each BGP multipath routing table entry will use additional memory. We recommend that you do not use this feature on a router with a low amount of available memory and especially if router is carries full Internet routing tables.

Route Reflector Limitation

When multiple iBGP paths installed in a routing table, a route reflector will advertise only one paths (next hop). If a router is behind a route reflector, all routers that are connected to multihomed sites will not be advertised unless a different route distinguisher is configured for each VRF.

Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

- [Multipath Load Sharing Between eBGP and iBGP, page 414](#)
- [eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network, page 415](#)
- [eBGP and iBGP Multipath Load Sharing With Route Reflectors, page 416](#)
- [Benefits of Multipath Load Sharing for Both eBGP and iBGP, page 416](#)

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.

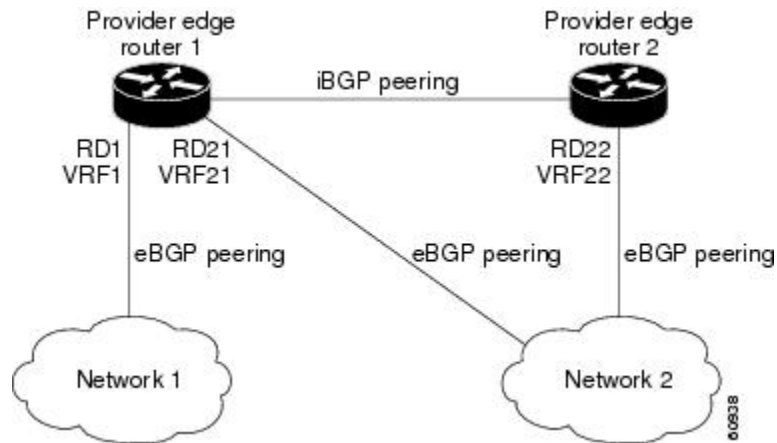
**Note**

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 40 A Service Provider BGP MPLS Network



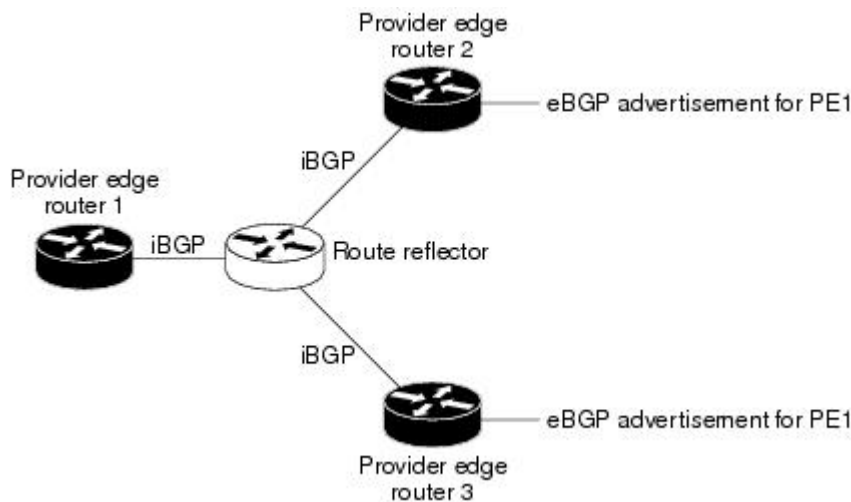
PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF of Network 1. The multipaths will be used by CEF to perform load balancing. IP traffic that is sent from Network 2 to PE router 1 and PE router 2 will be sent across the eBGP paths as IP traffic. IP traffic that is sent across the iBGP path will be sent as MPLS traffic, and MPLS traffic that is sent across an eBGP path will be sent as IP traffic. Any prefix that is advertised from Network 2 will be received by PE router 1 through route distinguisher (RD) 21 and RD 22. The advertisement through RD 21 will be carried in IP

packets, and the advertisement through RD 22 will be carried in MPLS packets. Both paths can be selected as multipaths for VRF1 and installed into the VRF1 RIB.

eBGP and iBGP Multipath Load Sharing With Route Reflectors

The figure below shows a topology that contains three PE routers and a route reflector, all configured for iBGP peering. PE router 2 and PE router 3 each advertise an equal preference eBGP path to PE router 1. By default, the route reflector will choose only one path and advertise PE router 1.

Figure 41 A Topology with a Route Reflector



For all equal preference paths to PE router 1 to be advertised through the route reflector, you must configure each VRF with a different RD. The prefixes received by the route reflector will be recognized differently and advertised to PE router 1.

Benefits of Multipath Load Sharing for Both eBGP and iBGP

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both eBGP and iBGP paths.

How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

- [Configuring Multipath Load Sharing for Both eBGP and iBGP](#), page 416
- [Verifying Multipath Load Sharing for Both eBGP and iBGP](#), page 418

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* [**import** *number*]
6. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
<p>Step 4 address-family ipv4 vrf <i>vrf-name</i></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4 vrf RED</pre>	<p>Places the router in address family configuration mode.</p> <ul style="list-style-type: none"> • Separate VRF multipath configurations are isolated by unique route distinguisher.
<p>Step 5 maximum-paths eibgp <i>number</i> [import <i>number</i>]</p> <p>Example:</p> <pre>Router(config-router-af)# maximum-paths eibgp 6</pre>	<p>Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table.</p> <p>Note The <code>maximum-paths eibgp</code> command can be configured only under the IPv4 VRF address family configuration mode and cannot be configured in any other address family configuration mode.</p>

Command or Action	Purpose
Step 6 <code>end</code> Example: <code>Router(config-router-af)# end</code>	Exits address family configuration mode, and enters Privileged EXEC mode.

Verifying Multipath Load Sharing for Both eBGP and iBGP

To verify this feature, perform the steps in this section

SUMMARY STEPS

1. `enable`
2. `show ip bgp neighbors` [*neighbor-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths**[*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 <code>enable</code> Example: <code>Router> enable</code>	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 <code>show ip bgp neighbors</code> [<i>neighbor-address</i> [advertised-routes dampened-routes flap-statistics paths [<i>regex</i>] received <i>prefix-filter</i> received-routes routes]] Example: <code>Router# show ip bgp neighbors</code>	Displays information about the TCP and BGP connections to neighbors.
Step 3 <code>show ip bgp vpnv4</code> { all rd <i>route-distinguisher</i> vrf <i>vrf-name</i> } Example: <code>Router# show ip bgp vpnv4 vrf RED</code>	Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP.

Command or Action	Purpose
Step 4 <code>show ip route vrf vrf-name</code> Example: Router# <code>show ip route vrf RED</code>	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.

Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature

- [eBGP and iBGP Multipath Load Sharing Configuration Example, page 419](#)
- [eBGP and iBGP Multipath Load Sharing Verification Examples, page 419](#)

eBGP and iBGP Multipath Load Sharing Configuration Example

This following configuration example configures a router in address-family mode to select six BGP routes (eBGP or iBGP) as multipaths:

```
Router(config)# router bgp 40000
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# maximum-paths eibgp 6
Router(config-router-af)# end
```

eBGP and iBGP Multipath Load Sharing Verification Examples

To verify that iBGP and eBGP routes have been configured for load sharing, use the `show ip bgp vpnv4 EXEC` command or the `show ip route vrf EXEC` command.

In the following example, the `show ip bgp vpnv4` command is entered to display multipaths installed in the VPNv4 RIB:

```
Router# show ip bgp vpnv4 all 10.22.22.0
BGP routing table entry for 10:1:22.22.22.0/24, version 19
Paths:(5 available, best #5)
Multipath:eiBGP
  Advertised to non peer-group peers:
  10.0.0.2 10.0.0.3 10.0.0.4 10.0.0.5
  22
    10.0.0.2 (metric 20) from 10.0.0.4 (10.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.4
    22
    10.0.0.2 (metric 20) from 10.0.0.5 (10.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.5
    22
    10.0.0.2 (metric 20) from 10.0.0.2 (10.0.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1 0x0:0:0
    22
    10.0.0.2 (metric 20) from 10.0.0.3 (10.0.0.3)
```

```

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:

```

Router# show ip route vrf PATH 10.22.22.0
Routing entry for 10.22.22.0/24
  Known via "bgp 1", distance 20, metric 0
  Tag 22, type external
  Last update from 10.1.1.12 01:59:31 ago
  Routing Descriptor Blocks:
  * 10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.4, 01:59:31 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.5, 01:59:31 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
  10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.2, 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" document.

Additional References

For additional information related to BGP Multipath Load sharing for Both eBGP and iBGP in an MPLS VPN, refer to the following references:

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Comprehensive BGP link bandwidth configuration examples and tasks	"BGP Link Bandwidth" module
CEF configuration tasks	"Cisco Express Forwarding Overview" module

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	A Border Gateway Protocol 4 (BGP4)
RFC 2547	BGP/MPLS VPNs
RFC 2858	Multiprotocol Extensions for BGP-4

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/techsupport

Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

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

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

Table 23 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 3.1.0SG	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. The following command was introduced or modified by this feature: maximum-paths eibgp .

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: 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)

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



Loadsharing IP Packets Over More Than Six Parallel Paths

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 423](#)
- [Overview of Loadsharing IP Packets over More Than Six Parallel Paths, page 423](#)
- [Additional References, page 424](#)
- [Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths, page 425](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Overview of Loadsharing IP Packets over More Than Six Parallel Paths

The Loadsharing IP Packets over More Than Six Parallel Paths feature increases the maximum number of parallel routes that can be installed to the routing table. The maximum number has been increased from six to sixteen for the following commands:

- **maximum-paths**
- **maximum-paths eibgp**
- **maximum-paths ibgp**

The output of the **show ip route summary** command has been updated to display the number of parallel routes supported by the routing table.

The benefits of this feature include the following:

- More flexible configuration of parallel routes in the routing table.
- Ability to configure multipath loadsharing over more links to allow for the configuration of higher-bandwidth aggregation using lower-speed links.

Additional References

For additional information related to multipath loadsharing and the configuration of parallel routes, see the following references:

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
eiBGP 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
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

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

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
<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/techsupport</p>

Feature Information for Loadsharing IP Packets Over More Than Six Parallel Paths

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

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

Table 24 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 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>The following commands were modified:</p> <ul style="list-style-type: none"> • maximum-paths • maximum-paths eibgp • maximum-paths ibgp • show ip route summary

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: 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)

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



BGP Policy Accounting

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 427](#)
- [Prerequisites, page 427](#)
- [Information About BGP Policy Accounting, page 427](#)
- [How to Configure BGP Policy Accounting, page 429](#)
- [Configuration Examples, page 432](#)
- [Additional References, page 433](#)
- [Feature Information for BGP Policy Accounting, page 434](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites

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, page 427](#)
- [Benefits of BGP Policy Accounting, page 428](#)

BGP Policy Accounting Overview

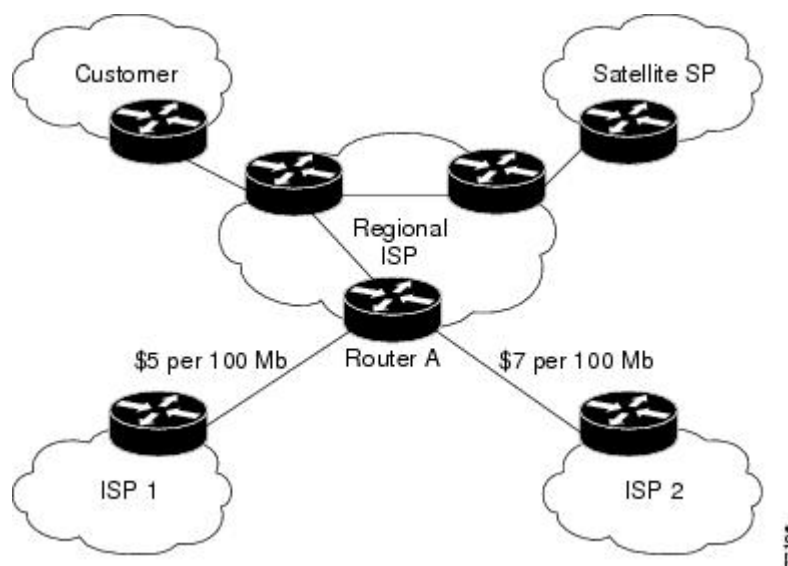
Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and counters based on

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 42 Sample Topology for BGP Policy Accounting



BGP policy accounting using autonomous system numbers can be used to improve the design of network circuit peering and transit agreements between Internet service providers (ISPs).

Benefits of BGP 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](#), page 429
- [Classifying the IP Traffic and Enabling BGP Policy Accounting](#), page 430
- [Verifying BGP Policy Accounting](#), page 431
- [Monitoring and Maintaining BGP Policy Accounting](#), page 432

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. Router(config)# **ip community-list** *community-list-number*{**permit**|**deny**} *community-number*
2. Router(config)# **route-map** *map-name*{**permit**|**deny**} [*sequence-number*]
3. Router(config-route-map)# **match community-list** *community-list-number*{**exact**}
4. Router(config-route-map)# **set traffic-index** *bucket-number*

DETAILED STEPS

Command or Action	Purpose
Step 1 Router(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 Router(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. 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 Router(config-route-map)# match community-list <i>community-list-number</i> { exact }	Matches a BGP community.
Step 4 Router(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. Router(config)# **router bgp** *as-number*
2. Router(config-router)# **table-map** *route-map-name*
3. Router(config-router)# **network** *network-number*[**mask** *network-mask*]
4. Router(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
5. Router(config-router)# **exit**
6. Router(config)# **interface** *interface-type* *interface-number*
7. Router(config-if)# **no ip directed-broadcast**
8. Router(config-if)# **ip address** *ip-address* *mask*
9. Router(config-if)# **bgp-policy accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# router bgp <i>as-number</i>	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 2	Router(config-router)# table-map <i>route-map-name</i>	Classifies BGP prefixes entered in the routing table.
Step 3	Router(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	Router(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	Router(config-router)# exit	Exits to global configuration mode.
Step 6	Router(config)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interface type and number and enters interface configuration mode.
Step 7	Router(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	Router(config-if)# ip address <i>ip-address</i> <i>mask</i>	Configures the interface with an IP address.
Step 9	Router(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:

```
Router# show ip cef 192.168.5.0 detail
192.168.5.0/24, version 21, cached adjacency to POS7/2
0 packets, 0 bytes, traffic_index 4
  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:

```
Router# show ip bgp 192.168.5.0
BGP routing table entry for 192.168.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    10.14.1.1 from 10.14.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:197
```

Step 3 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:

```
LC-Slot7# 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

To monitor and maintain the BGP Policy Accounting feature, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show cef interface [<i>type number</i>] policy-statistics	Displays detailed CEF policy statistical information for all interfaces.
Router# show ip bgp [<i>network</i>] [<i>network mask</i>] [longer-prefixes]	Displays entries in the BGP routing table.
Router# show ip cef [<i>network</i> [<i>mask</i>]] [detail]	Displays entries in the Forwarding Information Base (FIB) or FIB summary information.

Configuration Examples

- [Specifying the Match Criteria for BGP Policy Accounting Example, page 432](#)
- [Classifying the IP Traffic and Enabling BGP Policy Accounting Example, page 433](#)

Specifying the Match Criteria for BGP Policy Accounting Example

In the following example, BGP communities are specified in community lists, and a route map named `set_bucket` is configured to match each of the community lists to a specific accounting bucket using the `set traffic-index` command:

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
!
route-map set_bucket permit 10
match community 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

```

Classifying the IP Traffic and Enabling BGP Policy Accounting Example

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 Commands List, All Releases
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) commands	<i>Cisco IOS IP Switching Command Reference</i>
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) configuration information	"Cisco Express Forwarding Overview" module of the Cisco IOS Switching Services Configuration Guide

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> <li data-bbox="363 342 889 373">• CISCO-BGP-POLICY-ACCOUNTING-MIB <p data-bbox="342 394 889 548">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 data-bbox="922 331 1487 457">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 data-bbox="922 468 1252 506">http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
<p data-bbox="342 716 889 1005">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.</p>	<p data-bbox="922 716 1365 779">http://www.cisco.com/cisco/web/support/index.html</p>

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 25 **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	<p data-bbox="1154 348 1507 688">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 data-bbox="1154 711 1487 768">The following commands were introduced or modified:</p> <ul data-bbox="1154 791 1487 995" style="list-style-type: none"> <li data-bbox="1154 791 1325 819">• bgp-policy <li data-bbox="1154 829 1385 856">• set traffic-index <li data-bbox="1154 867 1487 924">• show cef interface policy-statistics <li data-bbox="1154 934 1341 961">• show ip bgp <li data-bbox="1154 972 1333 999">• show ip cef

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: 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)

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



BGP 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 437](#)
- [Prerequisites for the BGP Cost Community Feature, page 437](#)
- [Restrictions for the BGP Cost Community Feature, page 437](#)
- [Information About the BGP Cost Community Feature, page 438](#)
- [How to Configure the BGP Cost Community Feature, page 441](#)
- [Configuration Examples for the BGP Cost Community Feature, page 444](#)
- [Where to Go Next, page 445](#)
- [Additional References, page 445](#)
- [Command Reference, page 447](#)
- [Feature Information for BGP Cost Community, page 447](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for the BGP Cost Community Feature

This document assumes that BGP is configured in your network and that peering has been established.

Restrictions for the BGP Cost Community Feature

- The BGP Cost Community feature can be configured only within an autonomous system or confederation. The cost community is a non-transitive extended community that is passed to iBGP and confederation peers only and is not passed to eBGP peers.
- The BGP Cost Community feature must be supported on all routers in the autonomous system or confederation before cost community filtering is configured. The cost community should be applied consistently throughout the local autonomous system or confederation to avoid potential routing loops.
- Multiple cost community set clauses may be configured with the **set extcommunity cost** command in a single route map block or sequence. However, each set clause must be configured with a different ID value (0-255) for each point of insertion (POI). The ID value determines preference when all other attributes are equal. The lowest ID value is preferred.

Information About the BGP Cost Community Feature

- [BGP Cost Community Overview, page 438](#)
- [How the BGP Cost Community Influences the Best Path Selection Process, page 438](#)
- [Cost Community Support for Aggregate Routes and Multipaths, page 439](#)
- [Influencing Route Preference in a Multi-Exit IGP Network, page 440](#)
- [BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links, page 440](#)

BGP Cost Community Overview

The cost community is a non-transitive extended community attribute that is passed to iBGP and confederation peers but not to eBGP peers. The configuration of the BGP Cost Community feature allows you to customize the BGP best path selection process for a local autonomous system or confederation.

The cost community attribute is applied to internal routes by configuring the **set extcommunity cost** command in a route map. The cost community set clause is configured with a cost community ID number (0-255) and cost number (0-4294967295). The cost number value determines the preference for the path. The path with the lowest cost community number is preferred. Paths that are not specifically configured with the cost community attribute are assigned a default cost number value of 2147483647 (The midpoint between 0 and 4294967295) and evaluated by the best path selection process accordingly. In the case where two paths have been configured with the same cost number value, the path selection process will then prefer the path with the lowest cost community ID. The cost extended community attribute is propagated to iBGP peers when extended community exchange is enabled with the **neighbor send-community** command.

The following commands can be used to apply the route map that is configured with the cost community set clause:

- aggregate-address
- neighbor default-originate route-map {in | out}
- neighbor route-map
- network route-map
- redistribute route-map

How the BGP Cost Community Influences the Best Path Selection Process

The cost community attribute influences the BGP best path selection process at the point of insertion (POI). By default, the POI follows the IGP metric comparison. When BGP receives multiple paths to the same

destination, it uses the best path selection process to determine which path is the best path. BGP automatically makes the decision and installs the best path into the routing table. The POI allows you to assign a preference to a specific path when multiple equal cost paths are available. If the POI is not valid for local best path selection, the cost community attribute is silently ignored.

Multiple paths can be configured with the cost community attribute for the same POI. The path with the lowest cost community ID is considered first. In other words, all of the cost community paths for a specific POI are considered, starting with the one with the lowest cost community. Paths that do not contain the cost community (for the POI and community ID being evaluated) are assigned the default community cost value (2147483647). If the cost community values are equal, then cost community comparison proceeds to the next lowest community ID for this POI.

**Note**

Paths that are not configured with the cost community attribute are considered by the best path selection process to have the default cost-value (half of the maximum value [4294967295] or 2147483647).

Applying the cost community attribute at the POI allows you to assign a value to a path originated or learned by a peer in any part of the local autonomous system or confederation. The cost community can be used as a "tie breaker" during the best path selection process. Multiple instances of the cost community can be configured for separate equal cost paths within the same autonomous system or confederation. For example, a lower cost community value can be applied to a specific exit path in a network with multiple equal cost exits points, and the specific exit path will be preferred by the BGP best path selection process. See the scenario described in "Influencing Route Preference in a Multi-Exit IGP Network".

Cost Community Support for Aggregate Routes and Multipaths

Aggregate routes and multipaths are supported by the BGP Cost Community feature. The cost community attribute can be applied to either type of route. The cost community attribute is passed to the aggregate or multipath route from component routes that carry the cost community attribute. Only unique IDs are passed, and only the highest cost of any individual component route will be applied to the aggregate on a per-ID basis. If multiple component routes contain the same ID, the highest configured cost is applied to the route. For example, the following two component routes are configured with the cost community attribute via an inbound route map:

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

If these component routes are aggregated or configured as a multipath, the cost value 200 (POI=IGP, ID=1, Cost=200) will be advertised because it is the highest cost.

If one or more component routes does not carry the cost community attribute or if the component routes are configured with different IDs, then the default value (2147483647) will be advertised for the aggregate or multipath route. For example, the following three component routes are configured with the cost community attribute via an inbound route map. However, the component routes are configured with two different IDs.

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 172.16.0.1 (POI=IGP, ID=2, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

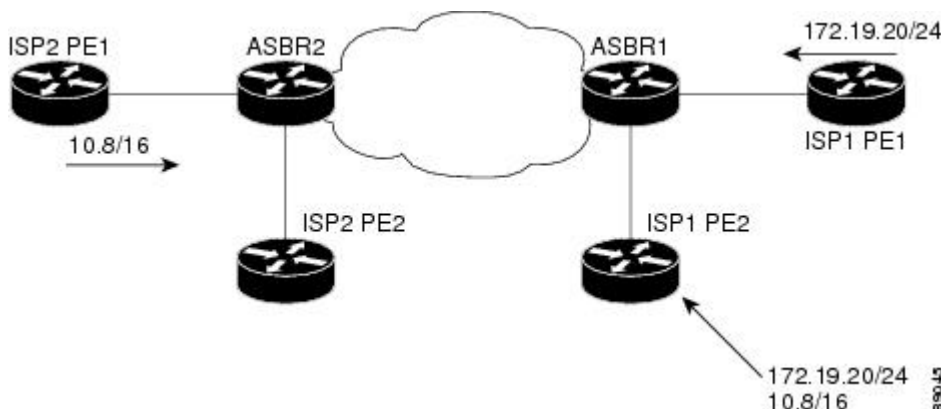
The single advertised path will include the aggregated cost communities as follows:

- {POI=IGP, ID=1, Cost=2147483647} {POI=IGP, ID=2, Cost=2147483647}

Influencing Route Preference in a Multi-Exit IGP Network

The figure below shows an Interior Gateway Protocol (IGP) network with two autonomous system boundary routers (ASBRs) on the edge. Each ASBR has an equal cost path to network 10.8/16.

Figure 43 Multi-Exit Point IGP Network



Both paths are considered to be equal by BGP. If multipath loadsharing is configured, both paths will be installed to the routing table and will be used to load balance traffic. If multipath load balancing is not configured, then BGP will select the path that was learned first as the best path and install this path to the routing table. This behavior may not be desirable under some conditions. For example, the path is learned from ISP1 PE2 first, but the link between ISP1 PE2 and ASBR1 is a low-speed link.

The configuration of the cost community attribute can be used to influence the BGP best path selection process by applying a lower cost community value to the path learned by ASBR2. For example, the following configuration is applied to ASBR2.

```
route-map ISP2_PE1 permit 10
  set extcommunity cost 1 1
  match ip address 13
!
ip access-list 13 permit 10.8.0.0 0.0.255.255
```

The above route map applies a cost community number value of 1 to the 10.8.0.0 route. By default, the path learned from ASBR1 will be assigned a cost community value of 2147483647. Because the path learned from ASBR2 has lower cost community value, this path will be preferred.

BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links

Before EIGRP Site of Origin (SoO) BGP Cost Community support was introduced, BGP preferred locally sourced routes over routes learned from BGP peers. Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. (A back door link, or a route, is a connection that is configured outside of the VPN between a remote and main site. For example, a WAN leased line that connects a remote site to the corporate network).

The "pre-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, page 441](#)
- [Verifying the Configuration of the BGP Cost Community, page 443](#)

Configuring the BGP Cost Community

To configure the cost community, perform the task in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **ipv6** [**multicast** | **unicast**] | **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	Enables higher privilege levels, such as privileged EXEC mode.
	Example:	<ul style="list-style-type: none"> • Enter your password if prompted.
	<code>Router> enable</code>	

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode to create or configure a BGP routing process.</p>
<p>Step 4 <code>neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 remote-as 101</pre>	<p>Establishes peering with the specified neighbor or peer-group.</p>
<p>Step 5 <code>address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] ipv6 [multicast unicast] vpnv4 [unicast]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family ipv4</pre>	<p>Places the router in address family configuration mode.</p>
<p>Step 6 <code>neighbor <i>ip-address</i> route-map <i>map-name</i> {in out}</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.0.0.1 route- map MAP-NAME in</pre>	<p>Applies an incoming or outgoing route map for the specified neighbor or peer-group.</p>
<p>Step 7 <code>exit</code></p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	<p>Exits router configuration mode and enters global configuration mode.</p>

	Command or Action	Purpose
Step 8	<p>route-map <i>map-name</i> {permit deny} [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Router(config)# route-map MAP-NAME permit 10</pre>	Enters route map configuration mode to create or configure a route map.
Step 9	<p>set extcommunity cost [igp] <i>community-id cost-value</i></p> <p>Example:</p> <pre>Router(config-route-map)# set extcommunity cost 1 100</pre>	<p>Creates a set clause to apply the cost community attribute.</p> <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>Router(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 "Verifying the Configuration of the BGP Cost Community" for specific example output.

- [Troubleshooting Tips, page 443](#)

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

- [BGP Cost Community Configuration Example, page 444](#)
- [BGP Cost Community Verification Examples, page 444](#)

BGP Cost Community Configuration Example

The following example configuration shows the configuration of the **set extcommunity cost** command. The following example applies the cost community ID of 1 and cost community value of 100 to routes that are permitted by the route map. This configuration will cause the best path selection process to prefer this route over other equal cost paths that were not permitted by this route map sequence.

```
Router(config)# router bgp 50000
Router(config-router)# neighbor 10.0.0.1 remote-as 50000
Router(config-router)# neighbor 10.0.0.1 update-source Loopback 0
Router(config-router)# address-family ipv4
Router(config-router-af)# neighbor 10.0.0.1 activate
Router(config-router-af)# neighbor 10.0.0.1 route-map COST1 in
Router(config-router-af)# neighbor 10.0.0.1 send-community both
Router(config-router-af)# exit
Router(config)# route-map COST1 permit 10
Router(config-route-map)# match ip-address 1
Router(config-route-map)# set extcommunity cost 1 100
```

BGP Cost Community Verification Examples

BGP cost community configuration can be verified locally or for a specific neighbor. To verify the local configuration cost community, use the **show route-map** or **show running-config** command. To verify that a specific neighbor carries the cost community, use the **show ip bgp ip-address** command.

The output of the **show route-map** command will display locally configured route-maps, match, set, continue clauses, and the status and configuration of the cost community attribute. The following sample output is similar to the output that will be displayed:

```
Router# show route-map
route-map COST1, permit, sequence 10
  Match clauses:
    as-path (as-path filter): 1
  Set clauses:
    extended community Cost:igp:1:100
  Policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 20
  Match clauses:
    ip next-hop (access-lists): 2
  Set clauses:
    extended community Cost:igp:2:200
  Policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 30
  Match clauses:
    interface FastEthernet0/0
    extcommunity (extcommunity-list filter):300
  Set clauses:
    extended community Cost:igp:3:300
  Policy routing matches: 0 packets, 0 bytes
```

The following sample output shows locally configured routes with large cost community values:

```
Router# show route-map
route-map set-cost, permit, sequence 10
  Match clauses:
```

```

Set clauses:
  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:

```

Router# show running-config | begin route-map
route-map COST1 permit 20
  match ip next-hop 2
  set extcommunity cost igp 2 200
!
route-map COST1 permit 30
  match interface FastEthernet0/0
  match extcommunity 300
  set extcommunity cost igp 3 300
.
.
.

```

The output of the **show ip bgp ip-address** command can be used to verify if a specific neighbor carries a path that is configured with the cost community attribute. The cost community attribute information is displayed in the "Extended Community" field. The POI, the cost community ID, and the cost community number value are displayed. The following sample output shows that neighbor 172.16.1.2 carries a cost community with an ID of 1 and a cost of 100:

```

Router# show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  2 2 2
    172.16.1.2 from 172.16.1.2 (172.16.1.2)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Extended Community: Cost:igp:1:100

```

If the specified neighbor is configured with the default cost community number value or if the default value is assigned automatically for cost community evaluation, "default" with + and - values will be displayed after the cost community number value in the output.

Where to Go Next

For more information about the EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature, refer to the "EIGRP MPLS VPN PE-CE Site of Origin (SoO)" module .

Additional References

For additional information related to the BGP Cost Community feature, refer to the following references:

Related Documents

Related Topic	Document Title
BGP Best Path Selection	"BGP Best Path Selection Algorithm"
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</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
draft-retana-bgp-custom-decision-00.txt	BGP Custom Decision Process

Technical Assistance

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Routing: BGP Command Reference*. For information about all Cisco IOS commands, go to the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or to the *Cisco IOS Master Commands List*.

- **bgp bestpath cost-community ignore**
- **debug ip bgp updates**
- **set extcommunity cost**

Feature Information for BGP Cost Community

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

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

Table 26 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>

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: 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)

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



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

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map.

- [Finding Feature Information, page 449](#)
- [Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 449](#)
- [Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 450](#)
- [Information About BGP Support for IP Prefix Import from Global Table into a VRF Table, page 450](#)
- [How to Import IP Prefixes from Global Table into a VRF Table, page 451](#)
- [Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 457](#)
- [Additional References, page 459](#)
- [Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table, page 460](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Border Gateway Protocol (BGP) peering sessions are established.
- CEF or dCEF (for distributed platforms) is enabled on all participating routers.

Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Only IPv4 unicast and multicast prefixes can be imported into a VRF with this feature.
- A maximum of five VRF instances per router can be created to import IPv4 prefixes from the global routing table.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a VPNv4 VRF.

Information About BGP Support for IP Prefix Import from Global Table into a VRF Table

- [Importing IPv4 Prefixes into a VRF, page 450](#)
- [Black Hole Routing, page 450](#)
- [Classifying Global Traffic, page 450](#)
- [Unicast Reverse Path Forwarding, page 451](#)

Importing IPv4 Prefixes into a VRF

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding instance (VRF) table using an import route map. This feature extends the functionality of VRF import-map configuration to allow IPv4 prefixes to be imported into a VRF based on a standard community. Both IPv4 unicast and multicast prefixes are supported. No Multiprotocol Label Switching (MPLS) or route target (import/export) configuration is required.

IP prefixes are defined as match criteria for the import map through standard Cisco filtering mechanisms. For example, an IP access-list, an IP prefix-list, or an IP as-path filter is created to define an IP prefix or IP prefix range, and then the prefix or prefixes are processed through a match clause in a route map. Prefixes that pass through the route map are imported into the specified VRF per the import map configuration.

Black Hole Routing

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be configured to support Black Hole Routing (BHR). BHR is a method that allows the administrator to block undesirable traffic, such as traffic from illegal sources or traffic generated by a Denial of Service (DoS) attack, by dynamically routing the traffic to a dead interface or to a host designed to collect information for investigation, mitigating the impact of the attack on the network. Prefixes are looked up, and packets that come from unauthorized sources are blackholed by the ASIC at line rate.

Classifying Global Traffic

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be used to classify global IP traffic based on physical location or class of service. Traffic is classified based on administration policy and then imported into different VRFs. On a college campus, for example, network traffic could be divided into an academic network and residence network traffic, a student network and faculty network, or

a dedicated network for multicast traffic. After the traffic is divided along administration policy, routing decisions can be configured with the MPLS VPN--VRF Selection Using Policy Based Routing feature or the MPLS VPN--VRF Selection Based on Source IP Address feature.

Unicast Reverse Path Forwarding

Unicast Reverse Path Forwarding (Unicast RPF) can be optionally configured with the BGP Support for IP Prefix Import from Global Table into a VRF Table feature. Unicast RPF is used to verify that the source address is in the Forwarding Information Base (FIB). The **ip verify unicast vrf** command is configured in interface configuration mode and is enabled for each VRF. This command has **permit** and **deny** keywords that are used to determine if the traffic is forwarded or dropped after Unicast RPF verification.

How to Import IP Prefixes from Global Table into a VRF Table

- [Defining IPv4 IP Prefixes to Import, page 451](#)
- [Creating the VRF and the Import Route Map, page 452](#)
- [Filtering on the Ingress Interface, page 455](#)
- [Verifying Global IP Prefix Import, page 456](#)

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 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
<p>Step 3 <code>access-list access-list-number {deny permit} source [source-wildcard] [log]</code></p> <p>Example:</p> <pre>Router(config)# access-list 50 permit 10.1.1.0 0.0.0.255</pre>	<p>Creates an access list and defines a range of IP prefixes to import into the VRF table.</p> <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.
<p>Step 4 <code>ip prefix-list prefix-list-name [seq seq-value] {deny network / length permit network / length} [ge ge-value] [le le-value]</code></p> <p>Example:</p> <pre>Router(config)# ip prefix-list COLORADO permit 10.24.240.0/22</pre>	<p>Creates a prefix list and defines a range of IP prefixes to import into the VRF table.</p> <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. 1000 prefixes per VRF are imported by default. You can manually configure from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you manually configure the prefix import limit. Configuring the router to import too many prefixes can interrupt normal router operation.

No MPLS or route target (import/export) configuration is required.

Import actions are triggered when a new routing update is received or when routes are withdrawn. During the initial BGP update period, the import action is postponed to allow BGP to converge 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. <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 vrf <i>vrf-name</i> Example: Router(config)# ip vrf GREEN	Creates a VRF routing table and specifies the VRF name (or tag). <ul style="list-style-type: none"> • The ip vrf <i>vrf-name</i> command creates a VRF routing table and a CEF table, and both are named using the <i>vrf-name</i> argument. Associated with these tables is the default route distinguisher value.

Command or Action	Purpose
<p>Step 4 <code>rd route-distinguisher</code></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).
<p>Step 5 <code>import ipv4 {unicast multicast} [prefix-limit] map route-map</code></p> <p>Example:</p> <pre>Router(config-vrf)# import ipv4 unicast 1000 map UNICAST</pre>	<p>Creates an import map to import IPv4 prefixes from the global routing table to a VRF table.</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 route-map that defines the prefixes to import is specified after the map keyword is entered. The example creates an import map that will import up to 1000 unicast prefixes that pass through the route map named UNICAST.
<p>Step 6 <code>exit</code></p> <p>Example:</p> <pre>Router(config-vrf)# exit</pre>	<p>Exits VRF configuration mode and enters global configuration mode.</p>
<p>Step 7 <code>route-map map-tag [permit deny] [sequence-number]</code></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.
<p>Step 8 <code>match ip address {acl-number [acl-number acl-name] acl-name [acl-name acl-number] prefix-list prefix-list-name [prefix-list-name]}</code></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.
<p>Step 9 <code>end</code></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 <code>ip verify unicast vrf <i>vrf-name</i> {deny permit}</code> 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 <code>end</code> 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 <code>enable</code> Enables privileged EXEC mode. Enter your password if prompted. Example: <pre>Router# enable</pre>
Step 2 <code>show ip bgp vpnv4 {all rd <i>route-distinguisher</i> vrf <i>vrf-name</i>}</code> 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>Router# show ip bgp vpnv4 all BGP table version is 15, local router ID is 10.1.1.1 Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S Stale Origin codes: i - IGP, e - EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path</pre>

```

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:

```

Router# show ip vrf detail
VRF academic; default RD 100:10; default VPNID <not set>
VRF Table ID = 1
  No interfaces
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:10
  Import VPN route-target communities
    RT:100:10
  Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
  No export route-map

```

Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table

- [Configuring Global IP Prefix Import Example, page 457](#)
- [Verifying Global IP Prefix Import Example, page 458](#)

Configuring Global IP Prefix Import Example

The following example imports unicast prefixes into the VRF named *green* using an IP prefix list and a route map:

This example starts in global configuration mode:

```
!
```

```

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

```

Verifying Global IP Prefix Import Example

The **show ip vrf** command or the **show ip bgp vpnv4** command can be used to verify that prefixes are imported from the global routing table to the VRF table.

The following example from the **show ip vrf** command shows the import route map named UNICAST is importing IPv4 unicast prefixes and the prefix import limit is 1000:

```

Router# show ip vrf detail
VRF green; default RD 200:1; default VPNID <not set>
  Interfaces:
    Se2/0
VRF Table ID = 1
  Export VPN route-target communities
    RT:200:10
  Import VPN route-target communities
    RT:200:10
  Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
VRF red; default RD 200:2; default VPNID <not set>
  Interfaces:
    Se3/0
VRF Table ID = 2
  Export VPN route-target communities
    RT:200:20
  Import VPN route-target communities
    RT:200:20
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix

```

The following example from the **show ip bgp vpnv4** command shows the import route map names, the prefix import limit and the actual number of imported prefixes, and the individual import entries:

```

Router# show ip bgp vpnv4 all

BGP table version is 18, local router ID is 10.131.127.252
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf green)
Import Map: UNICAST, Address-Family: IPv4 Unicast, Pfx Count/Limit: 1/1000
*>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
BGP commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
MPLS Layer 3 VPN configuration tasks	"Configuring MPLS Layer 3 VPNs"
VRF selection using policy based routing	"Directing MPLS VPN Traffic Using Policy-Based Routing"
VRF selection based on source IP address	"MPLS VPN-- VRF Selection Based on Source IP Address"

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 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
<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/techsupport

Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table

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

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

Table 27 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, ip verify unicast vrf.</p>

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: 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)

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



BGP per Neighbor SoO Configuration

The BGP per Neighbor SoO Configuration feature simplifies the configuration of the site-of-origin (SoO) value. 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 463](#)
- [Prerequisites for BGP per Neighbor SoO Configuration, page 463](#)
- [Restrictions for BGP per Neighbor SoO Configuration, page 463](#)
- [Information About Configuring BGP per Neighbor SoO, page 464](#)
- [How to Configure BGP per Neighbor SoO, page 466](#)
- [Configuration Examples for BGP per Neighbor SoO Configuration, page 477](#)
- [Where to Go Next, page 479](#)
- [Additional References, page 480](#)
- [Feature Information for BGP per Neighbor SoO Configuration, page 481](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP per Neighbor SoO Configuration

This feature assumes that a Border Gateway Protocol (BGP) network is configured and that Cisco Express Forwarding is enabled in your network.

Restrictions for BGP per Neighbor SoO Configuration

A BGP neighbor or peer policy template-based SoO configuration takes precedence over the SoO value configured in an inbound route map.

Information About Configuring BGP per Neighbor SoO

- [Site of Origin BGP Community Attribute](#), page 464
- [Route Distinguisher](#), page 464
- [BGP Support for 4-Byte Autonomous System Numbers](#), page 464
- [BGP per Neighbor Site of Origin Configuration](#), page 465
- [Benefits of BGP per Neighbor Site of Origin](#), page 465

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 Support for 4-Byte Autonomous System Numbers

In Cisco IOS Release 12.4(24)T, support was added for 4-byte autonomous system numbers as described in RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*. In Cisco IOS Release 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses asdot notation--1.2 for example--as the only configuration format, regular expression match, and output display, with no asplain support.

In Cisco IOS Release 12.2(33)SRE, 12.2(33)XNE, and later releases, BGP support for 4-octet (4-byte) autonomous system numbers using the asplain format as the default format was introduced. The default asplain format uses decimal value numbers such as 65536, but you can configure 4-byte autonomous system numbers in both the asplain and asdot 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.

For configuration examples involving 4-byte autonomous system numbers, see the [Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template Example](#), page 478 or the [Configuring a per Neighbor SoO Value Using a BGP neighbor Command and 4-Byte Autonomous System Numbers Example](#), page 479.

For more details about the Cisco implementation of BGP autonomous system number formats, see the the "Cisco BGP Overview" module.

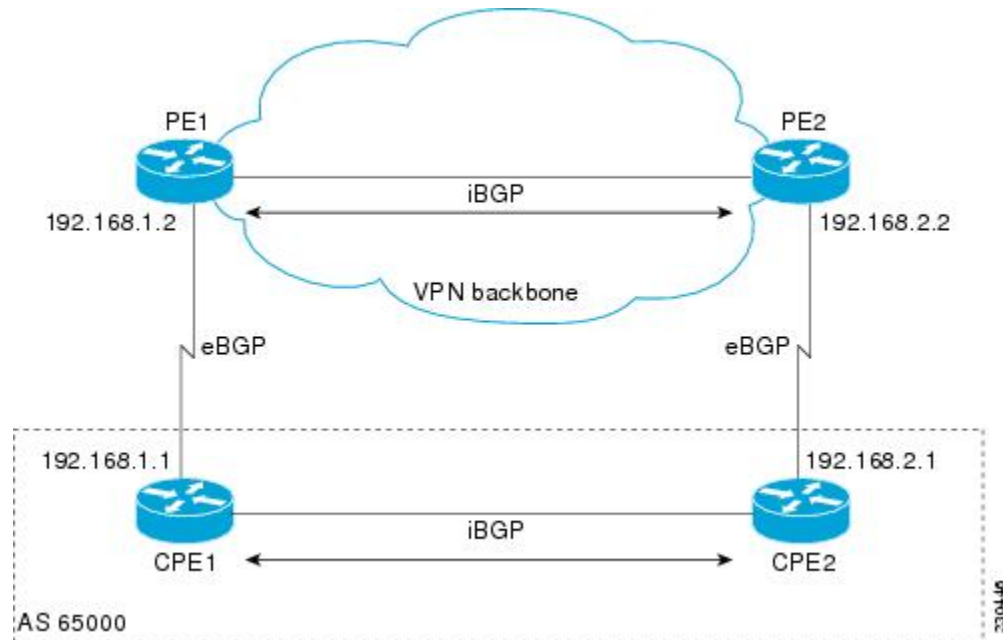
BGP per Neighbor Site of Origin Configuration

There are three ways to configure an SoO value for a BGP neighbor:

- BGP peer policy template--A peer policy template is created, and an SoO value is configured as part of the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.
- BGP **neighbor** command--Under address family IPv4 VRF, a neighbor is identified, and an SoO value is configured for the neighbor.
- BGP peer group--Under address family IPv4 VRF, a BGP peer group is configured, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

The configuration of SoO values for BGP neighbors is performed on a provider edge (PE) router, which is the VPN entry point. When SoO is enabled, the PE router forwards prefixes to the customer premises equipment (CPE) only when the SoO tag of the prefix does not match the SoO tag configured for the CPE. For example, in the figure below, an SoO tag is set as 65000:1 for the customer site that includes routers CPE1 and CPE2 with an autonomous system number of 65000. When CPE1 sends prefixes to PE1, PE1 tags the prefixes with 65000:1, which is the SoO tag for CPE1 and CPE2. When PE1 sends the tagged prefixes to PE2, PE2 performs a match against the SoO tag from CPE2. Any prefixes with the tag value of 65000:1 are not sent to CPE2 because the SoO tag matches the SoO tag of CPE2, and a routing loop is avoided.

Figure 44 Network Diagram for SoO Example



Benefits of BGP per Neighbor Site of Origin

In releases prior to Cisco IOS Release 12.4(11)T, 12.2(33)SRB, and 12.2(33)SB, the SoO extended community attribute is configured using an inbound route map that sets the SoO value during the update

process. The introduction of two new commands configured in submodes under router configuration mode simplifies the SoO value configuration.

How to Configure BGP per Neighbor SoO

To configure an SoO value for a BGP neighbor, you must perform the first task in the following list and one of the next three tasks. The last three tasks are mutually exclusive; you need perform only one of them.

- [Enabling Cisco Express Forwarding and Configuring VRF Instances, page 466](#)
- [Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template, page 469](#)
- [Configuring a per Neighbor SoO Value Using a BGP neighbor Command, page 472](#)
- [Configuring a per Neighbor SoO Value Using a BGP Peer Group, page 474](#)

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	<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>ip cef</p> <p>Example:</p> <pre>Router(config)# ip cef</pre>	<p>Enables Cisco Express Forwarding on the route processor.</p>
Step 4	<p>ip vrf vrf-name</p> <p>Example:</p> <pre>Router(config)# ip vrf SOO_VRF</pre>	<p>Defines a VRF instance and enters VRF configuration mode.</p>
Step 5	<p>rd route-distinguisher</p> <p>Example:</p> <pre>Router(config-vrf)# rd 1:1</pre>	<p>Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN.</p> <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.

Command or Action	Purpose
<p>Step 6 <code>route-target {export both} route-target-ext-community</code></p> <p>Example:</p> <pre>Router(config-vrf)# route-target export 1:1</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 <code>route-target-ext-community</code> 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>
<p>Step 7 <code>route-target {import both} route-target-ext-community</code></p> <p>Example:</p> <pre>Router(config-vrf)# route-target import 1:1</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 both keyword to both import routing information from, and export routing information to, the target VPN extended community. Use the <code>route-target-ext-community</code> argument to specify the VPN extended community.
<p>Step 8 <code>exit</code></p> <p>Example:</p> <pre>Router(config-vrf)# exit</pre>	<p>Exits VRF configuration mode and returns to global configuration mode.</p>
<p>Step 9 <code>interface type number</code></p> <p>Example:</p> <pre>Router(config)# interface GigabitEthernet 1/0/0</pre>	<p>Configures an interface type and enters interface configuration mode.</p>
<p>Step 10 <code>ip vrf forwarding vrf-name [downstream vrf-name2]</code></p> <p>Example:</p> <pre>Router(config-if)# ip vrf forwarding SOO_VRF</pre>	<p>Associates a VRF with an interface or subinterface.</p> <ul style="list-style-type: none"> In this example, the VRF named SOO_VRF is associated with Gigabit Ethernet interface 1/0/0. <p>Note Executing this command on an interface removes the IP address, so the IP address should be reconfigured.</p>
<p>Step 11 <code>ip address ip-address mask [secondary]</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 192.168.1.2 255.255.255.0</pre>	<p>Configures an IP address.</p> <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.

Command or Action	Purpose
Step 12 <code>end</code> Example: <pre>Router(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13 <code>show ip vrf [brief detail interfaces id] [vrf-name] [output-modifiers]</code> Example: <pre>Router# show ip vrf</pre>	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.

```
Router# show ip vrf
Name                Default RD          Interfaces
SOO_VRF             1:1                GE1/0/0
```

Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template

Perform this task on router PE1 in the figure above to configure an SoO value for a BGP neighbor at the router CPE1 in the figure above using a peer policy template. In this task, a peer policy template is created, and the SoO value is configured for the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.

For a configuration example involving 4-byte autonomous system numbers, see the [Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template Example, page 478](#).



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, page 469](#)

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.

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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

Command or Action	Purpose
<p>Step 3 <code>router bgp <i>autonomous-system-number</i></code></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 <code>template peer-policy <i>policy-template-name</i></code></p> <p>Example:</p> <pre>Router(config-router)# template peer-policy SOO_POLICY</pre>	<p>Creates a peer policy template and enters policy-template configuration mode.</p>
<p>Step 5 <code>soo <i>extended-community-value</i></code></p> <p>Example:</p> <pre>Router(config-router-ptmp)# soo 65000:1</pre>	<p>Sets the SoO value for a BGP peer policy template.</p> <ul style="list-style-type: none"> • Use the <i>extended-community-value</i> argument to specify the VPN extended community value. The value takes one of the following formats: <ul style="list-style-type: none"> ◦ A 16-bit autonomous system number, a colon, and a 32-bit number, for example: 45000:3 ◦ 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.
<p>Step 6 <code>exit-peer-policy</code></p> <p>Example:</p> <pre>Router(config-router-ptmp)# exit-peer-policy</pre>	<p>Exits policy-template configuration mode and returns to router configuration mode.</p>
<p>Step 7 <code>address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>]</code></p> <p>Example:</p> <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.

Command or Action	Purpose
<p>Step 8 <code>neighbor ip-address remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.1.1 remote-as 65000</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>
<p>Step 9 <code>neighbor ip-address activate</code></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 VRF address family with the local router.</p>
<p>Step 10 <code>neighbor ip-address inherit peer-policy policy-template-name</code></p> <p>Example:</p> <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.
<p>Step 11 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Configuring a per Neighbor SoO Value Using a BGP neighbor Command

Perform this task on router PE2 in the figure above to configure an SoO value for the BGP neighbor at router CPE2 in the figure above using a **neighbor** command. Under address family IPv4 VRF, a neighbor is identified, and an SoO value is configured for the neighbor.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer policy template configurations of the SoO value.

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
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 50000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 address-family ipv4 [unicast multicast] vrf <i>vrf-name</i>]</p> <p>Example:</p> <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.

Command or Action	Purpose
<p>Step 5 <code>neighbor {ip-address peer-group-name} remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.1 remote-as 65000</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>
<p>Step 6 <code>neighbor ip-address activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.</p> <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>
<p>Step 7 <code>neighbor {ip-address peer-group-name} soo extended-community-value</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 192.168.2.1 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 neighbor at 192.168.2.1 is configured with an SoO value of 65000:1.
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Configuring a per Neighbor SoO Value Using a BGP Peer Group

Perform this task on router PE1 in the figure above to configure an SoO value for the BGP neighbor at router CPE1 in the figure above using a **neighbor** command with a BGP peer group. Under address family IPv4 VRF, a BGP peer group is created and an SoO value is configured using a BGP **neighbor** command, and a neighbor is then identified and added as a peer group member. A BGP peer group member inherits the configuration associated with a peer group, which in this example, includes the SoO value.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer group configurations of the SoO value.

This task assumes that the task described in "Enabling Cisco Express Forwarding and Configuring VRF Instances" has been performed.

**Note**

A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**] **vrf** *vrf-name*
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** {*ip-address*|*peer-group-name*} **soo** *extended-community-value*
7. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
8. **neighbor** *ip-address* **activate**
9. **neighbor** *ip-address* **peer-group** *peer-group-name*
10. **end**

DETAILED STEPS

	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 50000	Enters router configuration mode for the specified routing process.

Command or Action	Purpose
<p>Step 4 address-family ipv4 [unicast multicast vrf <i>vrf-name</i>]</p> <p>Example:</p> <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.
<p>Step 5 neighbor <i>peer-group-name</i> peer-group</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor SOO_group peer-group</pre>	<p>Creates a BGP peer group.</p>
<p>Step 6 neighbor {<i>ip-address</i> <i>peer-group-name</i>} soo <i>extended-community-value</i></p> <p>Example:</p> <pre>Router(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.
<p>Step 7 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 65000</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>
<p>Step 8 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 VRF address family with the local router.</p>

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

Configuration Examples for BGP per Neighbor SoO Configuration

- [Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template Example, page 477](#)
- [Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template Example, page 478](#)
- [Configuring a per Neighbor SoO Value Using a BGP neighbor Command Example, page 478](#)
- [Configuring a per Neighbor SoO Value Using a BGP neighbor Command and 4-Byte Autonomous System Numbers Example, page 479](#)
- [Configuring a per Neighbor SoO Value Using a BGP Peer Group Example, page 479](#)

Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template Example

The following example shows how to create a peer policy template and configure an SoO value as part of the peer policy. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a peer policy template is created and an SoO value is configured as part of the peer policy. Under address family IPv4 VRF, a neighbor is identified and configured to inherit the peer policy that contains the SoO value.

```
ip cef
ip vrf SOO_VRF
 rd 1:1
  route-target export 1:1
  route-target import 1:1
 exit
interface GigabitEthernet 1/0/0
 ip vrf forwarding SOO_VRF
 ip address 192.168.1.2 255.255.255.0
 exit
router bgp 50000
 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

```

Configuring a per Neighbor SoO Value with a 4-Byte AS Number Using a BGP Peer Policy Template Example

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 address family IPv4 VRF, a neighbor is identified and configured to inherit the peer policy that contains the SoO value. This example requires Cisco IOS Release 12.4(24)T, or a later release.

```

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 address family IPv4 VRF, a neighbor is identified and configured to inherit the peer policy that contains the SoO value. This example requires Cisco IOS Release 12.2(33)SRE, 12.2(33)XNE, or a later release.

```

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
neighbor 192.168.3.2 activate
neighbor 192.168.3.2 inherit peer-policy SOO_POLICY
end

```

Configuring a per Neighbor SoO Value Using a BGP neighbor Command Example

The following example shows how to configure an SoO value for a BGP neighbor. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a neighbor is identified under address family IPv4 VRF and an SoO value is configured for the neighbor.

```

ip cef
ip vrf SOO_VRF
rd 1:1
route-target export 1:1
route-target import 1:1
exit
interface GigabitEthernet 1/0/0
ip vrf forwarding SOO_VRF
ip address 192.168.2.2 255.255.255.0
exit
router bgp 50000
address-family ipv4 vrf SOO_VRF
neighbor 192.168.2.1 remote-as 65000
neighbor 192.168.2.1 activate
neighbor 192.168.2.1 soo 65000:1
end

```

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

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 identified under address family IPv4 VRF, and an SoO value is configured for the neighbor. This example requires Cisco IOS Release 12.4(24)T, or a later release.

```
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
```

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

The following example shows how to configure an SoO value for a BGP peer group. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a BGP peer group is configured under address family IPv4 VRF, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

```
ip cef
ip vrf SOO_VRF
  rd 1:1
  route-target export 1:1
  route-target import 1:1
  exit
interface GigabitEthernet 1/0/0
  ip vrf forwarding SOO_VRF
  ip address 192.168.1.2 255.255.255.0
  exit
router bgp 50000
  address-family ipv4 vrf SOO_VRF
  neighbor SOO_GROUP peer-group
  neighbor SOO_GROUP soo 65000:65
  neighbor 192.168.1.1 remote-as 65000
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 peer-group SOO_GROUP
  end
```

Where to Go Next

- To read an overview of BGP, proceed to the "Cisco BGP Overview" module.
- To perform basic BGP feature tasks, proceed to the "Configuring a Basic BGP Network" module.
- To perform 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 perform internal BGP tasks, proceed to the "Configuring Internal BGP Features" module.

Additional References

The following sections provide references related to the BGP support for per neighbor SoO configuration feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
IP Switching commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Switching Command Reference</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 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.	http://www.cisco.com/cisco/web/support/index.html
To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.	
Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	

Feature Information for 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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 28 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(11)T 15.0(1)SY	<p>The BGP per neighbor SOO configuration feature simplifies the configuration of the site-of-origin (SoO) parameter. In Cisco IOS Release 12.4(9)T, 12.2(33)SRA, 12.2(31)SB2, and previous 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, soo.</p>

Feature Name	Releases	Feature Information
BGP Support for 4-Byte ASN	12.0(32)S12 12.0(32)SY8 12.2(33)SRE 12.2(33)XNE 12.4(24)T	<p>The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.</p> <p>In Cisco IOS Release 12.0(32)SY8, 12.2(33)SRE, and 12.2(33)XNE, the Cisco implementation of 4-byte autonomous system numbers uses <code>asplain</code> as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the <code>asplain</code> format and the <code>asdot</code> format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to <code>asdot</code> format, use the <code>bgp asnotation dot</code> command.</p> <p>In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses <code>asdot</code> as the only configuration format, regular expression match, and output display, with no <code>asplain</code> support.</p> <p>The following commands were modified by this feature: <code>bgp asnotation dot</code>, <code>bgp confederation identifier</code>, <code>bgp confederation peers</code>, <code>clear ip bgp</code>, <code>ip bgp-community new-format</code>, <code>ip extcommunity-list</code>, <code>match source-protocol</code>, <code>neighbor local-as</code>, <code>neighbor remote-as</code>, <code>neighbor soo</code>, <code>redistribute (IP)</code>, <code>router bgp</code>, <code>set</code></p>

Feature Name	Releases	Feature Information
		as-path, set extcommunity, set origin, soo, and all show ip bgp commands that display an autonomous system number.

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: 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)

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



Per-VRF Assignment of BGP Router ID

The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing **bgp router-id** command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

- [Finding Feature Information, page 485](#)
- [Prerequisites for Per-VRF Assignment of BGP Router ID, page 485](#)
- [Information About Per-VRF Assignment of BGP Router ID, page 485](#)
- [How to Configure Per-VRF Assignment of BGP Router ID, page 486](#)
- [Configuration Examples for Per-VRF Assignment of BGP Router ID, page 503](#)
- [Additional References, page 509](#)
- [Command Reference, page 510](#)
- [Feature Information for Per-VRF Assignment of BGP Router ID, page 511](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Per-VRF Assignment of BGP Router ID

Before you configure this feature, Cisco Express Forwarding (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, page 486](#)
- [Per-VRF Router ID Assignment, page 486](#)
- [Route Distinguisher, page 464](#)

BGP Router ID

The BGP router identifier (ID) is a 4-byte field that is set to the highest IP address on the router. Loopback interface addresses are considered before physical interface addresses because loopback interfaces are more stable than physical interfaces. The BGP router ID is used in the BGP algorithm for determining the best path to a destination where the preference is for the BGP router with the lowest router ID. It is possible to manually configure the BGP router ID using the **bgp router-id** command to influence the best path algorithm.

Per-VRF Router ID Assignment

In Cisco IOS Release 12.2(31)SB2, 12.2(33)SRA, 12.2(33)SXH, 12.4(20)T, and later releases, support for configuring separate router IDs for each Virtual Private Network (VPN) routing/forwarding (VRF) instance was introduced. The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing **bgp router-id** command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

Route Distinguisher

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to change it into a globally unique VPN-IPv4 prefix. An RD can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

```
45000:3
```

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

```
192.168.10.15:1
```

How to Configure Per-VRF Assignment of BGP Router ID

There are two main ways to configure a BGP router ID for each separate VRF. To configure a per-VRF BGP router ID manually, you must perform the first three tasks listed below. To automatically assign a BGP router ID to each VRF, perform the first task and the fourth task.

- [Configuring VRF Instances, page 486](#)
- [Associating VRF Instances with Interfaces, page 488](#)
- [Manually Configuring a BGP Router ID per VRF, page 491](#)
- [Automatically Assigning a BGP Router ID per VRF, page 496](#)

Configuring VRF Instances

Perform this task to configure VRF instances to be used with the per-VRF assignment tasks. In this task, a VRF instance named `vrf_trans` is created. To make the VRF functional, a route distinguisher is created.

When the route distinguisher is created, the routing and forwarding tables are created for the VRF instance named `vrf_trans`.

This task assumes that you have 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
<p>Step 1 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.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 ip vrf <i>vrf-name</i></p> <p>Example:</p> <pre>Router(config)# ip vrf vrf_trans</pre>	<p>Defines a VRF instance and enters VRF configuration mode.</p>
<p>Step 4 rd <i>route-distinguisher</i></p> <p>Example:</p> <pre>Router(config-vrf)# rd 45000:2</pre>	<p>Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN.</p> <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.

Command or Action	Purpose
<p>Step 5 <code>route-target [import both] route-target-ext-community</code></p> <p>Example:</p> <pre>Router(config-vrf)# route-target import 55000:5</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 both keyword to both import routing information from and export routing information to the target VPN extended community. Use the <code>route-target-ext-community</code> argument to specify the VPN extended community.
<p>Step 6 <code>route-target [export both] route-target-ext-community</code></p> <p>Example:</p> <pre>Router(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 <code>route-target-ext-community</code> argument to specify the VPN extended community.
<p>Step 7 <code>exit</code></p> <p>Example:</p> <pre>Router(config-vrf)# exit</pre>	<p>Exits VRF configuration mode and returns to global configuration mode.</p>
<p>Step 8 Repeat Step 3 through Step 7 for each VRF to be defined.</p>	--

Associating VRF Instances with Interfaces

Perform this task to associate VRF instances with interfaces to be used with the per-VRF assignment tasks. In this task, a VRF instance named `vrf_trans` is associated with a serial interface.

Make a note of the IP addresses for any interface to which you want to associate a VRF instance because the **ip vrf forwarding** command removes the IP address. Step 8 allows you to reconfigure the IP address.

- This task assumes that you have CEF or dCEF enabled.
- This task assumes that VRF instances have been configured in the [Configuring VRF Instances, page 486](#).

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.

	Command or Action	Purpose
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 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 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 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 ip-address mask [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] [vrf-name] 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.

This task assumes that you have previously created the VRF instances and associated them with interfaces. For more details, see the [Configuring VRF Instances](#), page 486 and the [Associating VRF Instances with Interfaces](#), page 488.

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**[*ttl*]
17. **neighbor** {*ip-address*|*peer-group-name*} **activate**
18. **neighbor** *ip-address* **allowas-in** [*number*]
19. **no auto-summary**
20. **no synchronization**
21. **bgp router-id** {*ip-address*| **auto-assign**}
22. Repeat Step 11 to Step 21 to configure another VRF instance.
23. **end**
24. **show ip bgp vpn4** {**all**| **rd** *route-distinguisher*| **vrf** *vrf-name*}

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>no bgp default ipv4-unicast</p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the IPv4 unicast address family for the BGP routing process.</p> <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>bgp log-neighbor-changes</p> <p>Example:</p> <pre>Router(config-router)# bgp log-neighbor-changes</pre>	<p>Enables logging of BGP neighbor resets.</p>
Step 6	<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 192.168.1.1 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"> 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.

Command or Action	Purpose
<p>Step 7 neighbor {<i>ip-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 update-source loopback0</pre>	<p>Allows BGP sessions to use any operational interface for TCP connections.</p> <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.
<p>Step 8 address-family {ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] 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.
<p>Step 9 neighbor {<i>ip-address</i> <i>peer-group-name</i>} 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.
<p>Step 10 neighbor {<i>ip-address</i> <i>peer-group-name</i>} 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.
<p>Step 11 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>

Command or Action	Purpose
<p>Step 12 <code>address-family {ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] vpnv4 [unicast]}</code></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.
<p>Step 13 <code>redistribute connected</code></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>.
<p>Step 14 <code>neighbor {ip-address peer-group-name} remote-as autonomous-system-number</code></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.
<p>Step 15 <code>neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]</code></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.
<p>Step 16 <code>neighbor {ip-address peer-group-name} ebgp-multihop[ttl]</code></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.

Command or Action	Purpose
<p>Step 17 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.
<p>Step 18 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.
<p>Step 19 no auto-summary</p> <p>Example:</p> <pre>Router(config-router-af)# no auto- summary</pre>	<p>Disables automatic summarization and sends subprefix routing information across classful network boundaries.</p>
<p>Step 20 no synchronization</p> <p>Example:</p> <pre>Router(config-router-af)# no synchronization</pre>	<p>Enables the Cisco IOS software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).</p>
<p>Step 21 bgp router-id {<i>ip-address</i> auto-assign}</p> <p>Example:</p> <pre>Router(config-router-af)# bgp router- id 10.99.1.1</pre>	<p>Configures a fixed router ID for the local BGP routing process.</p> <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.
<p>Step 22 Repeat Step 11 to Step 21 to configure another VRF instance.</p>	<p>--</p>
<p>Step 23 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>

Command or Action	Purpose
<p>Step 24 <code>show ip bgp vpnv4 {all rd route-distinguisher vrf vrf-name}</code></p> <p>Example:</p> <pre>Router# show ip bgp vpnv4 all</pre>	<p>(Optional) Displays VPN address information from the BGP table.</p> <ul style="list-style-type: none"> In this example, the complete VPNv4 database is displayed. <p>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>.</p>

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.

This task assumes that you have previously created the VRF instances. For more details, see the [Configuring VRF Instances, page 486](#).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*]
8. **ip address** *ip-address mask* [**secondary**]
9. Repeat Step 5 through Step 8 for each VRF to be associated with an interface.
10. **exit**
11. **router bgp** *autonomous-system-number*
12. **bgp router-id** {*ip-address*| **vrf auto-assign**}
13. **no bgp default ipv4-unicast**
14. **bgp log-neighbor-changes**
15. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
16. **neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number*
17. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [*vrf vrf-name*] | *vrf vrf-name*] | **vpn4** [**unicast**]}
18. **neighbor** {*ip-address*| *peer-group-name*} **activate**
19. **neighbor** {*ip-address*| *peer-group-name*} **send-community** {**both**| **standard**| **extended**}
20. **exit-address-family**
21. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [*vrf vrf-name*] | *vrf vrf-name*] | **vpn4** [**unicast**]}
22. **redistribute connected**
23. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
24. **neighbor** *ip-address* **local-as** *autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]
25. **neighbor** {*ip-address*| *peer-group-name*} **ebgp-multihop**[*ttl*]
26. **neighbor** {*ip-address*| *peer-group-name*} **activate**
27. **neighbor** *ip-address* **allowas-in** [*number*]
28. **no auto-summary**
29. **no synchronization**
30. Repeat Step 20 to Step 29 to configure another VRF instance.
31. **end**
32. **show ip bgp vpn4** {**all**| **rd** *route-distinguisher*| *vrf vrf-name*}

DETAILED STEPS

	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>interface <i>type number</i></p> <p>Example:</p> <pre>Router(config)# interface loopback0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> In this example, loopback interface 0 is configured.
Step 4	<p>ip address <i>ip-address mask</i> [secondary]</p> <p>Example:</p> <pre>Router(config-if)# ip address 172.16.1.1 255.255.255.255</pre>	<p>Configures an IP address.</p> <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	<p>exit</p> <p>Example:</p> <pre>Router(config-if)# exit</pre>	<p>Exits interface configuration mode and returns to global configuration mode.</p>
Step 6	<p>interface <i>type number</i></p> <p>Example:</p> <pre>Router(config)# interface loopback1</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> In this example, loopback interface 1 is configured.
Step 7	<p>ip vrf forwarding <i>vrf-name</i> [downstream <i>vrf-name2</i>]</p> <p>Example:</p> <pre>Router(config-if)# ip vrf forwarding vrf_trans</pre>	<p>Associates a VRF with an interface or subinterface.</p> <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>

Command or Action	Purpose
<p>Step 8 <code>ip address ip-address mask [secondary]</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 10.99.1.1 255.255.255.255</pre>	<p>Configures an IP address.</p> <ul style="list-style-type: none"> In this example, loopback interface 1 is configured with an IP address of 10.99.1.1.
<p>Step 9 Repeat Step 5 through Step 8 for each VRF to be associated with an interface.</p>	<p>--</p>
<p>Step 10 <code>exit</code></p> <p>Example:</p> <pre>Router(config-if)# exit</pre>	<p>Exits interface configuration mode and returns to global configuration mode.</p>
<p>Step 11 <code>router bgp autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config)# router bgp 45000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 12 <code>bgp router-id {ip-address vrf auto-assign}</code></p> <p>Example:</p> <pre>Router(config-router)# bgp router-id vrf auto-assign</pre>	<p>Configures a fixed router ID for the local BGP routing process.</p> <ul style="list-style-type: none"> In this example, a BGP router ID is automatically assigned for each VRF instance.
<p>Step 13 <code>no bgp default ipv4-unicast</code></p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the IPv4 unicast address family for the BGP routing process.</p> <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>
<p>Step 14 <code>bgp log-neighbor-changes</code></p> <p>Example:</p> <pre>Router(config-router)# bgp log- neighbor-changes</pre>	<p>Enables logging of BGP neighbor resets.</p>

Command or Action	Purpose
<p>Step 15 <code>neighbor {ip-address peer-group-name}</code> <code>remote-as autonomous-system-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 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"> • 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.
<p>Step 16 <code>neighbor {ip-address peer-group-name}</code> <code>update-source interface-type interface-number</code></p> <p>Example:</p> <pre>Router(config-router)# neighbor 192.168.1.1 update-source loopback0</pre>	<p>Allows BGP sessions to use any operational interface for TCP connections.</p> <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.
<p>Step 17 <code>address-family {ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] vpnv4 [unicast]}</code></p> <p>Example:</p> <pre>Router(config-router)# address-family vpnv4</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.
<p>Step 18 <code>neighbor {ip-address peer-group-name}</code> <code>activate</code></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.
<p>Step 19 <code>neighbor {ip-address peer-group-name}</code> <code>send-community {both standard extended}</code></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.

Command or Action	Purpose
<p>Step 20 <code>exit-address-family</code></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>
<p>Step 21 <code>address-family {ipv4 [mdt multicast unicast [vrf vrf-name] vrf vrf-name] vpnv4 [unicast]}</code></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.
<p>Step 22 <code>redistribute connected</code></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>.
<p>Step 23 <code>neighbor {ip-address peer-group-name} remote-as autonomous-system-number</code></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.
<p>Step 24 <code>neighbor ip-address local-as autonomous-system-number [no-prepend [replace-as [dual-as]]]</code></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.

Command or Action	Purpose
<p>Step 25 <code>neighbor {ip-address peer-group-name} ebgp-multihop[ttl]</code></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.
<p>Step 26 <code>neighbor {ip-address peer-group-name} activate</code></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.
<p>Step 27 <code>neighbor ip-address allowas-in [number]</code></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.
<p>Step 28 <code>no auto-summary</code></p> <p>Example:</p> <pre>Router(config-router-af)# no auto- summary</pre>	<p>Disables automatic summarization and sends subprefix routing information across classful network boundaries.</p>
<p>Step 29 <code>no synchronization</code></p> <p>Example:</p> <pre>Router(config-router-af)# no synchronization</pre>	<p>Enables the Cisco IOS software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).</p>
<p>Step 30 Repeat Step 20 to Step 29 to configure another VRF instance.</p>	<p>--</p>
<p>Step 31 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>

Command or Action	Purpose
Step 32 <code>show ip bgp vpnv4 {all rd route-distinguisher vrf vrf-name}</code> 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

- [Manually Configuring a BGP Router ID per VRF Examples, page 503](#)
- [Automatically Assigning a BGP Router ID per VRF Examples, page 505](#)

Manually Configuring a BGP Router ID per VRF Examples

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. The BGP router ID for each VRF is configured manually under separate IPv4 address families. The `show ip bgp vpnv4` command can be used to verify that the router IDs have been configured for each VRF. The configuration starts in global configuration mode.

```
ip vrf vrf_trans
```

```

rd 45000:1
route-target export 50000:50
route-target import 40000:1
!
ip vrf vrf_user
rd 65500:1
route-target export 65500:1
route-target import 65500:1
!
interface Loopback0
ip address 10.1.1.1 255.255.255.255
!
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
redistribute connected
neighbor 172.23.1.1 remote-as 50000
neighbor 172.23.1.1 local-as 40000 no-prepend
neighbor 172.23.1.1 ebgp-multihop 2
neighbor 172.23.1.1 activate
neighbor 172.23.1.1 allowas-in 1
no auto-summary
no synchronization
bgp router-id 10.99.1.2
exit-address-family

```

After the configuration, the output of the **show ip bgp vpnv4 all** command shows the router ID displayed next to the VRF name:

```

Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 172.22.0.0        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 vpnv4 vrf** command for a specified VRF displays the router ID in the output header:

```

Router# show ip bgp vpnv4 vrf vrf_user
BGP table version is 43, local router ID is 10.99.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
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 vpnv4 vrf summary** command for a specified VRF displays the router ID in the first line of the output:

```

Router# show ip bgp vpnv4 vrf vrf_user summary
BGP router identifier 10.99.1.1, local AS number 45000
BGP table version is 43, main routing table version 43
8 network entries using 1128 bytes of memory
8 path entries using 544 bytes of memory
16/10 BGP path/bestpath attribute entries using 1856 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
3 BGP extended community entries using 72 bytes of memory
0 BGP 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 vpnv4 vrf** command for a specified VRF and network address:

```

Router# show ip bgp vpnv4 vrf vrf_user 172.23.0.0
BGP routing table entry for 65500:1:172.23.0.0/8, version 22
Paths: (1 available, best #1, table vrf_user)
  Advertised to update-groups:
    2
    3
  Local
    0.0.0.0 from 0.0.0.0 (10.99.1.1)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:65500:1

```

Automatically Assigning a BGP Router ID per VRF Examples

The following three configuration examples show different methods of configuring BGP to automatically assign a separate router ID to each VRF instance:

Globally Automatically Assigned Router ID Using Loopback Interface IP Addresses

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. Under router configuration mode, BGP is globally configured to automatically assign each VRF a BGP router ID. Loopback interfaces are associated with individual VRFs to source an IP address for the router ID. The **show ip bgp vpnv4** command can be used to verify that the router IDs have been configured for each VRF.

```
ip vrf vrf_trans
 rd 45000:1
 route-target export 50000:50
 route-target import 40000:1
!
ip vrf vrf_user
 rd 65500:1
 route-target export 65500:1
 route-target import 65500:1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
 ip vrf forwarding vrf_user
 ip address 10.99.1.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vrf_trans
 ip address 10.99.2.2 255.255.255.255
!
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
 bgp router-id vrf auto-assign
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.3.1 remote-as 45000
 neighbor 192.168.3.1 update-source Loopback0
!
address-family vpnv4
 neighbor 192.168.3.1 activate
 neighbor 192.168.3.1 send-community extended
 exit-address-family
!
address-family ipv4 vrf vrf_user
 redistribute connected
 neighbor 172.22.1.1 remote-as 40000
 neighbor 172.22.1.1 local-as 50000 no-prepend
 neighbor 172.22.1.1 ebgp-multihop 2
 neighbor 172.22.1.1 activate
 neighbor 172.22.1.1 allowas-in 1
 no auto-summary
 no synchronization
 exit-address-family
!
address-family ipv4 vrf vrf_trans
 redistribute connected
 neighbor 172.23.1.1 remote-as 50000
 neighbor 172.23.1.1 local-as 2 no-prepend
 neighbor 172.23.1.1 ebgp-multihop 2
 neighbor 172.23.1.1 activate
 neighbor 172.23.1.1 allowas-in 1
 no auto-summary
 no synchronization
 exit-address-family
```


After the configuration, the output of the **show ip bgp vpnv4 all** command shows the router ID displayed next to the VRF name. Note that the router IDs used in this example are sourced from the IP addresses configured for loopback interface 1 and loopback interface 2. The router IDs are the same as in the [Manually Configuring a BGP Router ID per VRF Examples, page 503](#).

```
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 vpnv4** command can be used to verify that the router IDs have been configured for each VRF.

```
ip vrf vrf_trans
  rd 45000:1
  route-target export 50000:50
  route-target import 40000:1
!
ip vrf vrf_user
  rd 65500:1
  route-target export 65500:1
  route-target import 65500:1
!
interface Loopback0
  ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
  ip vrf forwarding vrf_user
  ip address 10.99.1.1 255.255.255.255
!
interface Loopback2
  ip vrf forwarding vrf_trans
  ip address 10.99.2.2 255.255.255.255
!
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           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 ?

```

Additional References

The following sections provide references related to the Per-VRF Assignment of BGP Router ID feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
MPLS commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples	<i>Cisco IOS Multiprotocol Label Switching Command Reference</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 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Routing: BGP Command Reference*. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **bgp router-id**
- **show ip bgp vpnv4**

Feature Information for Per-VRF Assignment of BGP Router ID

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

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

Table 29 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 vpnv4.</p>

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: 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)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams,

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.



BGP Next Hop Unchanged

In an external BGP (eBGP) session, by default, the router changes the next hop attribute of a BGP route (to its own address) when the router sends out a route. The BGP Next Hop Unchanged feature allows BGP to send an update to an eBGP multihop peer with the next hop attribute unchanged.

- [Finding Feature Information, page 513](#)
- [Information About BGP Next Hop Unchanged, page 513](#)
- [How to Configure BGP Next Hop Unchanged, page 514](#)
- [Configuration Example for BGP Next Hop Unchanged, page 516](#)
- [Additional References, page 517](#)
- [Feature Information for BGP Next Hop Unchanged, page 517](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Information About BGP Next Hop Unchanged

- [BGP Next Hop Unchanged, page 513](#)

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, page 515](#)

Configuring the BGP Next Hop Unchanged for an eBGP Peer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **address-family { ipv4 [unicast] | vpnv4 [unicast] }**
5. **neighbor *ip-address* remote-as *as-number***
6. **neighbor *ip-address* activate**
7. **neighbor *ip-address* ebgp-multihop *ttl***
8. **neighbor *ip-address* next-hop-unchanged**
9. **end**
10. **show ip bgp**

DETAILED STEPS

	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>as-number</i> Example: Router(config)# router bgp 65535	Enters router configuration mode, and creates a BGP routing process.
Step 4	address-family { ipv4 [unicast] vpnv4 [unicast] } Example: Router(config-router-af)# address-family vpnv4	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.

	Command or Action	Purpose
Step 5	<p>neighbor ip-address remote-as as-number</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.100 remote-as 65600</pre>	Adds an entry to the BGP neighbor table.
Step 6	<p>neighbor ip-address activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.100 activate</pre>	Enables the exchange of information with the peer.
Step 7	<p>neighbor ip-address ebgp-multihop ttl</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255</pre>	Configures the local router to accept and initiate connections to external peers that reside on networks that are not directly connected.
Step 8	<p>neighbor ip-address next-hop-unchanged</p> <p>Example:</p> <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	<p>end</p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode, and enters privileged EXEC mode.
Step 10	<p>show ip bgp</p> <p>Example:</p> <pre>Router# show ip bgp</pre>	<p>(Optional) Displays entries in the BGP routing table.</p> <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, page 517](#)

Example BGP Next Hop Unchanged for an eBGP Peer

The following example configures a multihop eBGP peer at 10.0.0.100 in a remote AS. When the local router sends updates to that peer, it will send them without modifying the next hop attribute.

```
router bgp 65535
 address-family ipv4
  neighbor 10.0.0.100 remote-as 65600
  neighbor 10.0.0.100 activate
  neighbor 10.0.0.100 ebgp-multihop 255
  neighbor 10.0.0.100 next-hop-unchanged
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
BGP configuration tasks	<i>IP Routing: BGP Configuration Guide</i>
MPLS configuration tasks	<i>MPLS Configuration Guide</i>
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer	"Configuring Internal BGP Features"

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 30 **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 added by this feature: neighbor next-hop-unchanged .

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: 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)

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



BGP Support for the L2VPN Address Family

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

- [Finding Feature Information, page 519](#)
- [Prerequisites for BGP Support for the L2VPN Address Family, page 519](#)
- [Restrictions for BGP Support for the L2VPN Address Family, page 519](#)
- [Information About BGP Support for the L2VPN Address Family, page 520](#)
- [How to Configure BGP Support for the L2VPN Address Family, page 521](#)
- [Configuration Examples for BGP Support for the L2VPN Address Family, page 527](#)
- [Where to Go Next, page 530](#)
- [Additional References, page 530](#)
- [Feature Information for BGP Support for the L2VPN Address Family, page 531](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP Support for the L2VPN Address Family

The BGP Support for the L2VPN Address Family feature assumes prior knowledge of VPN, Virtual Private LAN Service (VPLS), and Multiprotocol Layer Switching (MPLS) technologies.

Restrictions for BGP Support for the L2VPN Address Family

- For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used in L2VPN address family configuration mode. All other route map commands are supported.
- BGP multipaths and confederations are not supported in the L2VPN address family.

Information About BGP Support for the L2VPN Address Family

- [L2VPN Address Family](#), page 520
- [VPLS ID](#), page 521

L2VPN Address Family

In Cisco IOS Release 12.2(33)SRB and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or generic routing encapsulation (GRE). The L2VPN address family is configured in BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

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

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

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

- **bgp nexthop**
- **bgp scan-time**
- **neighbor activate**
- **neighbor advertisement-interval**
- **neighbor allowas-in**
- **neighbor capability**
- **neighbor inherit**
- **neighbor maximum-prefix**
- **neighbor next-hop-self**
- **neighbor next-hop-unchanged**
- **neighbor peer-group**
- **neighbor remove-private-as**
- **neighbor route-map**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor soft-reconfiguration**

- **neighbor soo**
- **neighbor weight**

**Note**

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

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

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

VPLS ID

A VPLS ID is a BGP extended community value that identifies the VPLS domain. Manual configuration of this ID is optional because a default VPLS ID is generated using the BGP autonomous system number and the configured VPN ID. A VPLS ID can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter a VPLS ID in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

45000:3

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

192.168.10.15:1

How to Configure BGP Support for the L2VPN Address Family

- [Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family, page 521](#)

Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family

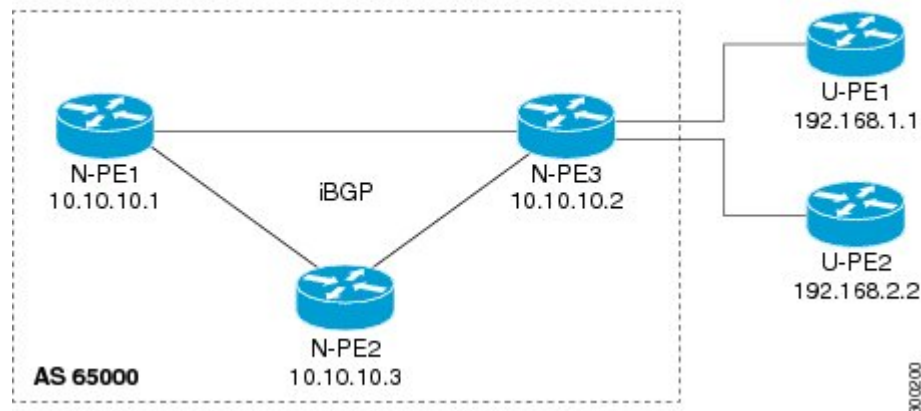
Perform this task to implement VPLS autodiscovery of each provider edge (PE) router that is a member of a specific VPLS. In Cisco IOS Release 12.2(33)SRB, the BGP L2VPN address family was introduced with a separate L2VPN RIB that contains endpoint provisioning information. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time any Layer 2 (L2) VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

BGP-based VPLS autodiscovery eliminates the need to manually provision a VPLS neighbor. After a PE router configures itself to be a member of a particular VPLS, information needed to set up connections to remote routers in the same VPLS is distributed by a discovery process. When the discovery process is complete, each member of the VPLS will have the information needed to set up VPLS pseudowires to form the full mesh of pseudowires needed for the VPLS.

This task is configured at router N-PE3 in the figure below and must be repeated at routers N-PE1 and N-PE2 with the appropriate changes such as different IP addresses. For a full configuration of these routers, see “Configuring VPLS Autodiscovery Using BGP” and the “L2VPN Address Family Example”.

Figure 45 Network Diagram for BGP Autodiscovery Using the L2VPN Address Family

N-PE - Network-facing PE router
U-PE - User-facing PE router



In this task, the PE router N-PE3 in the figure above is configured with a Layer 2 router ID, a VPN ID, a VPLS ID, and is enabled to automatically discover other PE routers that are part of the same VPLS domain. A BGP session is created to activate BGP neighbors in the L2VPN address family. Finally, two optional **show** commands are entered to verify the steps in the task.

If a route reflector (RR) node is provisioned for a new virtual forwarding instance (VFI), BGP announces the entire current table from the L2VPN address family identifier (AFI) to the L2VPN xconnect database to make sure that the virtual circuits (VC) are active.

This task assumes that MPLS is configured with VPLS options. For more details, see the VPLS Autodiscovery: BGP Based feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 router-id** *ip-address*
4. **l2 vfi** *vfi-name* **autodiscovery**
5. **vpn id** *vpn-id*
6. **vpls-id** *vpls-id*
7. **exit**
8. Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.
9. **router bgp** *autonomous-system-number*
10. **no bgp default ipv4-unicast**
11. **bgp log-neighbor-changes**
12. **bgp update-delay** *seconds*
13. **neighbor** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number*
14. **neighbor** { *ip-address* | *peer-group-name* } **update-source** *interface-type interface-number*
15. Repeat Step 13 and Step 14 to configure other BGP neighbors.
16. **address-family l2vpn** [**vpls**]
17. **neighbor** { *ip-address* | *peer-group-name* } **activate**
18. **neighbor** { *ip-address* | *peer-group-name* } **send-community** [**both** | **standard** | **extended**]
19. Repeat Step 17 and Step 18 to activate other BGP neighbors in L2VPN address family.
20. **end**
21. **show vfi**
22. **show ip bgp l2vpn vpls** { **all** | **rd** *vpn-rd* }

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.

Command or Action	Purpose
Step 3 <code>l2 router-id ip-address</code> Example: <pre>Router(config)# l2 router-id 10.1.1.3</pre>	Specifies a router ID (in IP address format) for the PE router to use with VPLS autodiscovery pseudowires. <ul style="list-style-type: none"> In this example, the L2 router ID is defined as 10.1.1.3.
Step 4 <code>l2 vfi vfi-name autodiscovery</code> Example: <pre>Router(config)# l2 vfi customerA autodiscovery</pre>	Creates an L2 VFI, enables the VPLS PE router to automatically discover other PE routers that are part of the same VPLS domain, and enters L2 VFI autodiscovery configuration mode. <ul style="list-style-type: none"> In this example, the L2 VFI named customerA is created.
Step 5 <code>vpn id vpn-id</code> Example: <pre>Router(config-vfi)# vpn id 100</pre>	Specifies a VPN ID. <ul style="list-style-type: none"> Use the same VPN ID for the PE routers that belong to the same VPN. Make sure that the VPN ID is unique for each VPN in the service provider network. Use the <i>vpn-id</i> argument to specify a number in the range from 1 to 4294967295. In this example, a VPN ID of 100 is specified.
Step 6 <code>vpls-id vpls-id</code> Example: <pre>Router(config-vfi)# vpls-id 65000:100</pre>	(Optional) Specifies a VPLS ID. <ul style="list-style-type: none"> The VPLS ID is an identifier that is used to identify the VPLS domain. This command is optional because a default VPLS ID is automatically generated using the BGP autonomous system number and the VPN ID configured for the VFI. Only one VPLS ID can be configured per VFI, and the same VPLS ID cannot be configured in multiple VFIs on the same router. In this example, a VPLS ID of 65000:100 is specified.
Step 7 <code>exit</code> Example: <pre>Router(config-vfi)# exit</pre>	Exits L2 VFI autodiscovery configuration mode and returns to global configuration mode.
Step 8 Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.	—
Step 9 <code>router bgp autonomous-system-number</code> Example: <pre>Router(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.

Command or Action	Purpose
<p>Step 10 no bgp default ipv4-unicast</p> <p>Example:</p> <pre>Router(config-router)# no bgp default ipv4-unicast</pre>	<p>Disables the IPv4 unicast address family for the BGP routing process.</p> <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>
<p>Step 11 bgp log-neighbor-changes</p> <p>Example:</p> <pre>Router(config-router)# bgp log- neighbor-changes</pre>	<p>Enables logging of BGP neighbor resets.</p>
<p>Step 12 bgp update-delay <i>seconds</i></p> <p>Example:</p> <pre>Router(config-router)# bgp update- delay 1</pre>	<p>Sets the maximum initial delay period before a BGP-speaking networking device sends its first updates.</p> <ul style="list-style-type: none"> Use the <i>seconds</i> argument to set the delay period.
<p>Step 13 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 10.10.10.1 remote-as 65000</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 10.10.10.1 is an internal BGP neighbor.
<p>Step 14 neighbor {<i>ip-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.10.10.1 update-source loopback 1</pre>	<p>(Optional) Configures a router to select a specific source or interface to receive routing table updates.</p> <ul style="list-style-type: none"> This example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface.
<p>Step 15 Repeat Step 13 and Step 14 to configure other BGP neighbors.</p>	<p>--</p>

Command or Action	Purpose
<p>Step 16 <code>address-family l2vpn [vpls]</code></p> <p>Example:</p> <pre>Router(config-router)# address-family l2vpn vpls</pre>	<p>Specifies the L2VPN address family and enters address family configuration mode.</p> <ul style="list-style-type: none"> The optional vpls keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers. In this example, an L2VPN VPLS address family session is created.
<p>Step 17 <code>neighbor {ip-address peer-group-name} activate</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.10.10.1 activate</pre>	<p>Enables the neighbor to exchange information for the L2VPN VPLS address family with the local router.</p> <p>Note If you have configured a BGP peer group as a neighbor, you do not use this step. BGP peer groups are activated when a BGP parameter is configured. For example, the neighbor send-community command in the next step will automatically activate a peer group.</p>
<p>Step 18 <code>neighbor {ip-address peer-group-name} send-community[both standard extended]</code></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.10.10.1 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 10.10.10.1.
<p>Step 19 Repeat Step 17 and Step 18 to activate other BGP neighbors in L2VPN address family.</p>	--
<p>Step 20 <code>end</code></p> <p>Example:</p> <pre>Router(config-router-af)# end</pre>	<p>Exits address family configuration mode and returns to privileged EXEC mode.</p>
<p>Step 21 <code>show vfi</code></p> <p>Example:</p> <pre>Router# show vfi</pre>	<p>(Optional) Displays information about the configured VFI instances.</p>
<p>Step 22 <code>show ip bgp l2vpn vpls {all rd vpn-rd}</code></p> <p>Example:</p> <pre>Router# show ip bgp l2vpn vpls all</pre>	<p>(Optional) Displays information about the L2 VPN VPLS address family.</p>

- [Examples, page 527](#)
- [What to Do Next, page 527](#)

Examples

The following is sample output from the **show vfi** command that shows two VFIs, CustomerA and CustomerB, with their associated VPN and VPLS IDs:

```
Router# show vfi
Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No
VFI name: customerA, state: down, type: multipoint
  VPN ID: 100, VPLS-ID: 65000:100
  RD: 65000:100, RT: 65000:100
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address      VC ID      Discovered Router ID  S
  10.10.10.1        100        10.10.10.99           Y
VFI name: customerB, state: down, type: multipoint
  VPN ID: 200, VPLS-ID: 65000:200
  RD: 65000:200, RT: 65000:200
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address      VC ID      Discovered Router ID  S
  10.10.10.3        200        10.10.10.98           Y
```

The following is sample output from the **show ip bgp l2vpn vpls all** command that shows two VFIs identified by their VPN route distinguisher:

```
Router# show ip bgp l2vpn vpls all
BGP table version is 5, local router ID is 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65000:100
*> 65000:100:10.10.10.1/96
           0.0.0.0                                32768 ?
*>i65000:100:192.168.1.1/96
           10.10.10.2                            0   100   0 ?
Route Distinguisher: 65000:200
*> 65000:200:10.10.10.3/96
           0.0.0.0                                32768 ?
*>i65000:200:192.168.2.2/96
           10.10.10.2                            0   100   0 ?
```

What to Do Next

To configure more VPLS features, see the main VPLS documentation in the "VPLS Autodiscovery: BGP Based" feature.

Configuration Examples for BGP Support for the L2VPN Address Family

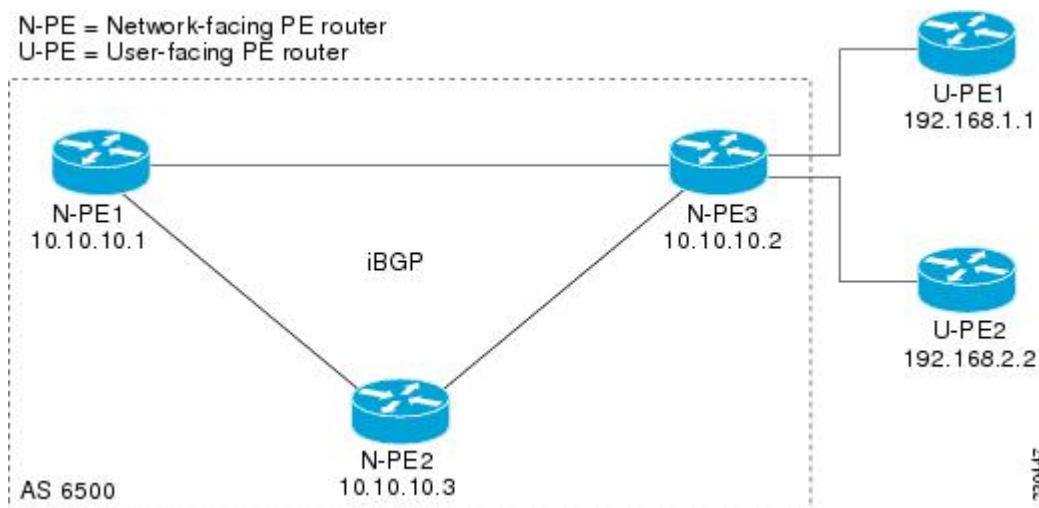
- [Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example, page 527](#)

Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example

In this configuration example, all the routers in autonomous system 65000 in the figure below are configured to provide BGP support for the L2VPN address family. VPLS autodiscovery is enabled and L2

VFI and VPN IDs are configured. BGP neighbors are configured and activated in L2VPN address family to ensure that the VPLS endpoint provisioning information is saved to a separate L2VPN RIB and then distributed to the other BGP peers in BGP update messages. When the endpoint information is received by the BGP peers, a pseudowire mesh is set up to support L2VPN-based services.

Figure 46 Network Diagram for VPLS Autodiscovery Using BGP and the L2VPN Address Family



Router N-PE1

```

ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 1000 2000
mpls label protocol ldp
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
  vpn id 100
!
pseudowire-class mpls
  encapsulation mpls
!
interface Loopback1
  ip address 10.1.1.1 255.255.255.255
!
interface Ethernet0/0
  description Backbone interface
  ip address 10.0.0.1 255.255.255.0
  mpls ip
!
router ospf 1
  log-adjacency-changes
  network 10.10.1.0 0.0.0.255 area 0
  network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  bgp update-delay 1
  neighbor 10.10.10.2 remote-as 65000
  neighbor 10.10.10.2 update-source Loopback 1
  neighbor 10.10.10.3 remote-as 65000
  neighbor 10.10.10.3 update-source Loopback 1
!
address-family l2vpn vpls
  
```

```
neighbor 10.10.10.2 activate
neighbor 10.10.10.2 send-community extended
neighbor 10.10.10.3 activate
neighbor 10.10.10.3 send-community extended
exit-address-family
!
ip classless
```

Router N-PE2

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
  vpn id 100
!
pseudowire-class mpls
  encapsulation mpls
!
interface Loopback1
  ip address 10.1.1.2 255.255.255.255
!
interface Ethernet0/0
  description Backbone interface
  ip address 10.0.0.2 255.255.255.0
  mpls ip
!
router ospf 1
  log-adjacency-changes
  network 10.10.1.0 0.0.0.255 area 0
  network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  bgp update-delay 1
  neighbor 10.10.10.1 remote-as 65000
  neighbor 10.10.10.1 update-source Loopback 1
  neighbor 10.10.10.3 remote-as 65000
  neighbor 10.10.10.3 update-source Loopback 1
!
  address-family l2vpn vpls
  neighbor 10.10.10.1 activate
  neighbor 10.10.10.1 send-community extended
  neighbor 10.10.10.3 activate
  neighbor 10.10.10.3 send-community extended
  exit-address-family
!
ip classless
```

Router N-PE3

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
  vpn id 100
!
pseudowire-class mpls
  encapsulation mpls
```

```

!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface Ethernet0/0
 description Backbone interface
 ip address 10.0.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.1 update-source Loopback 1
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.2 update-source Loopback 1
!
 address-family l2vpn vpls
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-community extended
 neighbor 10.10.10.2 activate
 neighbor 10.10.10.2 send-community extended
 exit-address-family
!
 ip classless

```

Where to Go Next

For more details about configuring VPLS autodiscovery, see the "VPLS Autodiscovery: BGP Based" feature.

Additional References

The following sections provide references related to the BGP Support for the L2VPN Address Family feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
BGP overview	"Cisco BGP Overview" <i>module</i>
Configuring basic BGP tasks	"Configuring a Basic BGP Network" <i>module</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 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for the L2VPN Address Family

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software

release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 31 Feature Information for BGP Support for the L2VPN Address Family

Feature Name	Releases	Feature Information
BGP Support for the L2VPN Address Family	12.2(33)SRB 15.1(1)SG Cisco IOS XE 3.3SG	<p>BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN RIB to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.</p> <p>In Cisco IOS Release 12.2(33)SRB, this feature was introduced on the Cisco 7600 platform.</p> <p>The following commands were introduced or modified by this feature: address-family l2vpn, clear ip bgp l2vpn, show ip bgp l2vpn.</p>

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: 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)

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



BGP 4 MIB Support for per-Peer Received Routes

This module describes the BGP 4 MIB Support for per-Peer Received Routes feature, introduces a new table in the CISCO-BGP4-MIB that provides the capability to query (by using Simple Network Management Protocol [SNMP] commands) for routes that are learned from individual Border Gateway Protocol (BGP) peers.

- [Finding Feature Information, page 533](#)
- [Restrictions on BGP 4 MIB Support for Per-Peer Received Routes, page 533](#)
- [Information About BGP 4 MIB Support for Per-Peer Received Routes, page 534](#)
- [Additional References, page 538](#)
- [Feature Information for BGP 4 MIB Support for per-Peer Received Routes, page 539](#)
- [Glossary, page 539](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Restrictions on BGP 4 MIB Support for Per-Peer Received Routes

BGP 4 MIB Support for per-Peer Received Routes supports only routes that are contained in IPv4 AFIs and unicast SAFIs in the local BGP RIB table. The BGP 4 MIB Support for per-Peer Received Routes enhancement is supported only by BGP Version 4.

Information About BGP 4 MIB Support for Per-Peer Received Routes

- [BGP 4 MIB Support for Per-Peer Received Routes Overview, page 534](#)
- [BGP 4 per-Peer Received Routes Table Elements and Objects, page 535](#)
- [Benefits of BGP 4 MIB Support for Per-Peer Received Routes, page 537](#)

BGP 4 MIB Support for Per-Peer Received Routes Overview

BGP 4 MIB Support for per-Peer Received Routes introduces a new table in the CISCO-BGP4-MIB that provides the capability to query (by using SNMP commands) for routes that are learned from individual BGP peers.

Before this new MIB table was introduced, a network operator could obtain the routes learned by a local BGP-speaking router by querying the local BGP speaker with an SNMP command (for example, the **snmpwalk** command). The network operator used the SNMP command to query the `bgp4PathAttrTable` of the CISCO-BGP4-MIB. The routes that were returned from a `bgp4PathAttrTable` query were indexed in the following order:

- Prefix
- Prefix length
- Peer address

Because the `bgp4PathAttrTable` indexes the prefixes first, obtaining routes learned from individual BGP peers will require the network operator to "walk through" the complete `bgp4PathAttrTable` and filter out routes from the interested peer. A BGP Routing Information Base (RIB) could contain 10,000 or more routes, which makes a manual "walk" operation impossible and automated walk operations very inefficient.

BGP 4 MIB Support for per-Peer Received Routes introduces a Cisco-specific enterprise extension to the CISCO-BGP4-MIB that defines a new table called the `cbgpRouterTable`. The `cbgpRouterTable` provides the same information as the `bgp4PathAttrTable` with the following two differences:

- Routes are indexed in the following order:
 - Peer address
 - Prefix
 - Prefix length

The search criteria for SNMP queries of local routes are improved because peer addresses are indexed before prefixes. A search for routes that are learned from individual peers is improved with this enhancement because peer addresses are indexed before prefixes. A network operator will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.

- Support is added for multiprotocol BGP, Address Family Identifier (AFI), and Subsequent Address Family Identifier (SAFI) information. This information is added in the form of indexes to the `cbgpRouterTable`. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.

**Note**

The MIB will be populated only if the router is configured to run a BGP process. The present implementation of BGP 4 MIB Support for per-Peer Received Routes will show only routes contained in IPv4 AFI and unicast SAFI BGP local RIB tables. Support for showing routes contained in other local RIB tables will be added in the future.

BGP 4 per-Peer Received Routes Table Elements and Objects

The following sections describe new table elements, AFI and SAFI tables and objects, and network address prefixes in the Network Layer Reachability Information (NLRI) fields that have been introduced by the BGP 4 MIB Support for per-Peer Received Routes enhancement.

- [MIB Tables and Objects, page 535](#)
- [AFIs and SAFIs, page 536](#)
- [Network Address Prefix Descriptions for the NLRI Field, page 536](#)

MIB Tables and Objects

The table below describes the MIB indexes of the `cbgpRouterTable`.

For a complete description of the MIB, see the CISCO-BGP4-MIB file CISCO-BGP4-MIB.my, available through Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

Table 32 **MIB Indexes of the `cbgpRouterTable`**

MIB Indexes	Description
<code>cbgpRouteAfi</code>	Represents the AFI of the network layer protocol that is associated with the route.
<code>cbgpRouteSafi</code>	Represents the SAFI of the route. It gives additional information about the type of the route. The AFI and SAFI are used together to determine which local RIB (Loc-RIB) contains a particular route.
<code>cbgpRoutePeerType</code>	Represents the type of network layer address that is stored in the <code>cbgpRoutePeer</code> object.
<code>cbgpRoutePeer</code>	Represents the network layer address of the peer from which the route information has been learned.
<code>cbgpRouteAddrPrefix</code>	Represents the network address prefix that is carried in a BGP update message. See the table below for information about the types of network layer addresses that can be stored in specific types of AFI and SAFI objects.

MIB Indexes	Description
cbgpRouteAddrPrefixLen	Represents the length in bits of the network address prefix in the NLRI field. See the table below for a description of the 13 possible entries.

AFIs and SAFIs

The table below lists the AFI and SAFI values that can be assigned to or held by the `cbgpRouteAfi` and `cbgpRouteSafi` indexes, respectively. The table below also displays the network address prefix type that can be held by specific combinations of AFIs and SAFIs. The type of network address prefix that can be carried in a BGP update message depends on the combination of AFIs and SAFIs.

Table 33 *AFIs and SAFIs*

AFI	SAFI	Type
ipv4(1)	unicast(1)	IPv4 address
ipv4(1)	multicast(2)	IPv4 address
ipv4(1)	vpn(128)	VPN-IPv4 address
ipv6(2)	unicast(1)	IPv6 address



Note

A VPN-IPv4 address is a 12-byte quantity that begins with an 8-byte Route Distinguisher (RD) and ends with a 4-byte IPv4 address. Any bits beyond the length specified by `cbgpRouteAddrPrefixLen` are represented as zeros.

Network Address Prefix Descriptions for the NLRI Field

The table below describes the length in bits of the network address prefix in the NLRI field of the `cbgpRouteTable`. Each entry in the table provides information about the route that is selected by any of the six indexes in the table below.

Table 34 *Network Address Prefix Descriptions for the NLRI Field*

Table or Object (or Index)	Description
<code>cbgpRouteOrigin</code>	The ultimate origin of the route information.
<code>cbgpRouteASPathSegment</code>	The sequence of autonomous system path segments.
<code>cbgpRouteNextHop</code>	The network layer address of the autonomous system border router that traffic should pass through to get to the destination network.

Table or Object (or Index)	Description
cbgpRouteMedPresent	Indicates that the MULTI_EXIT_DISC attribute for the route is either present or absent.
cbgpRouteMultiExitDisc	Metric that is used to discriminate between multiple exit points to an adjacent autonomous system. The value of this object is irrelevant if the value of the cbgpRouteMedPresent object is "false(2)."
cbgpRouteLocalPrefPresent	Indicates that the LOCAL_PREF attribute for the route is either present or absent.
cbgpRouteLocalPref	Determines the degree of preference for an advertised route by an originating BGP speaker. The value of this object is irrelevant if the value of the cbgpRouteLocalPrefPresent object is "false(2)."
cbgpRouteAtomicAggregate	Determines if the system has selected a less specific route without selecting a more specific route.
cbgpRouteAggregatorAS	The autonomous system number of the last BGP speaker that performed route aggregation. A value of 0 indicates the absence of this attribute.
cbgpRouteAggregatorAddrType	Represents the type of network layer address that is stored in the cbgpRouteAggregatorAddr object.
cbgpRouteAggregatorAddr	The network layer address of the last BGP 4 speaker that performed route aggregation. A value of all zeros indicates the absence of this attribute.
cbgpRouteBest	An indication of whether this route was chosen as the best BGP 4 route.
cbgpRouteUnknownAttr	One or more path attributes not understood by the local BGP speaker. A size of 0 indicates that this attribute is absent.

Benefits of BGP 4 MIB Support for Per-Peer Received Routes

- Improved SNMP Query Capabilities--The search criteria for SNMP queries for routes that are advertised by individual peers are improved because the peer address is indexed before the prefix. A network operator will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.
- Improved AFI and SAFI Support--Support is added for multiprotocol BGP. AFI and SAFI are added as indexes to the table. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.

Additional References

Related Documents

Related Topic	Document Title
Configuring MIBs for BGP	Configuring Advanced BGP Features
BGP commands	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Configuring SNMP Support	Configuring SNMP Support
SNMP Commands	"SNMP Commands" in <i>Cisco IOS Network Management Command Reference</i>

Standards

Standard	Title
None	--

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
RFC 1657	<i>BGP-4 MIB</i>
RFC 1771	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 2547	<i>BGP/MPLS VPNs</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</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>	<p>http://www.cisco.com/techsupport</p>

Feature Information for BGP 4 MIB Support for per-Peer Received Routes

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

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

Table 35 Feature Information for BGP 4 MIB Support for per-Peer Received Routes

Feature Name	Releases	Feature Configuration Information
BGP 4 MIB Support for per-Peer Received Routes	12.0(21)S 12.2(14)S 12.2(28)SB 15.0(1)S	<p>This feature introduces a new table in the CISCO-BGP4-MIB that provides the capability to query (by using SNMP commands) for routes that are learned from individual BGP peers.</p> <p>No commands were introduced or modified by this feature.</p>

Glossary

AFI--Address Family Identifier. Carries the identity of the network layer protocol that is associated with the network address.

BGP--Border Gateway Protocol. An interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163, A Border Gateway Protocol (BGP). The current

implementation of BGP is BGP Version 4 (BGP4). BGP4 is the predominant interdomain routing protocol that is used on the Internet. It supports CIDR and uses route aggregation mechanisms to reduce the size of routing tables.

MBGP--multiprotocol BGP. An enhanced version of BGP that carries routing information for multiple network layer protocols and IP multicast routes. It is defined in RFC 2858, Multiprotocol Extensions for BGP-4.

MIB--Management Information Base. A group of managed objects that are contained within a virtual information store or database. MIB objects are stored so that values can be assigned to object identifiers and to assist managed agents by defining which MIB objects should be implemented. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

NLRI--Network Layer Reachability Information. Carries route attributes that describe a route and how to connect to a destination. This information is carried in BGP update messages. A BGP update message can carry one or more NLRI prefixes.

RIB--Routing Information Base (RIB). A central repository of routes that contains Layer 3 reachability information and destination IP addresses or prefixes. The RIB is also known as the routing table.

SAFI--Subsequent Address Family Identifier. Provides additional information about the type of the Network Layer Reachability Information that is carried in the attribute.

SNMP--Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and to manage configurations, statistics collection, performance, and security.

snmpwalk --The **snmpwalk** command is a Simple Network Management Protocol (SNMP) application that is used to communicate with a network entity MIB using SNMP.

VPN--Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses a tunnel to encrypt all information at the IP level.

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: 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)

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



BGP 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 541](#)
- [Prerequisites for BGP Event-Based VPN Import, page 541](#)
- [Information About BGP Event-Based VPN Import, page 541](#)
- [How to Configure BGP Event-Based VPN Import, page 543](#)
- [Configuration Examples for BGP Event-Based VPN Import, page 549](#)
- [Where to Go Next, page 550](#)
- [Additional References, page 550](#)
- [Feature Information for BGP Event-Based VPN Import, page 551](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for BGP Event-Based VPN Import

Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Information About BGP Event-Based VPN Import

- [BGP Event-Based VPN Import, page 542](#)

BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing BGP path import process. BGP Virtual Private Network (VPN) import provides importing functionality for BGP paths where BGP paths are imported from the BGP VPN table into a BGP virtual routing and forwarding (VRF) topology. In the existing path import process, when path updates occur, the import updates are processed during the next scan time which is a configurable interval of 5 to 15 seconds. The scan time adds a delay in the propagation of routes. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available.

Using the BGP Event-Based VPN Import feature, convergence times are significantly reduced because provider edge (PE) routers can propagate VPN paths to customer edge (CE) routers without the scan time delay. Configuration changes such as adding imported route-targets to a VRF are not processed immediately, and are still handled during the 60-second periodic scanner pass.

- [Import Path Selection Policy, page 542](#)
- [Import Path Limit, page 542](#)

Import Path Selection Policy

Event-based VPN import introduces three path selection policies:

- **All--Import** all available paths from the exporting net that match any Route Target (RT) associated with the importing VRF instance.
- **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 bestpath, the next selection priority is for multipaths, and the lowest selection priority is for nonmultipaths.

How to Configure BGP Event-Based VPN Import

- [Configuring a Multiprotocol VRF](#), page 543
- [Configuring Event-Based VPN Import Processing for BGP Paths](#), page 546
- [Monitoring and Troubleshooting BGP Event-Based VPN Import Processing](#), page 547

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.
	Example:	
	Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>vrf definition vrf-name</code></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.
<p>Step 4 <code>rd route-distinguisher</code></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.
<p>Step 5 <code>route-target {import export both} route-target-ext-community</code></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.
<p>Step 6 <code>address-family ipv4 [unicast]</code></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.
<p>Step 7 <code>exit-address-family</code></p> <p>Example:</p> <pre>Router(config-vrf-af)# exit-address-family</pre>	<p>Exits VRF address family configuration mode and returns to VRF configuration mode.</p>

	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.

This task assumes that you have previously configured the VRF to be used with the VRF address family syntax. To configure a VRF, see the [Configuring a Multiprotocol VRF](#), page 543.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the [Configuring Event-Based VPN Import Processing for BGP Paths Example](#), page 549.

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. • 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
<p>Step 4 <code>address-family ipv4 vrf vrf-name</code></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.
<p>Step 5 <code>import path selection {all bestpath [strict] multipath [strict]}</code></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.
<p>Step 6 <code>import path limit number-of-import-paths</code></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>
<p>Step 7 <code>end</code></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

- [Configuring Event-Based VPN Import Processing for BGP Paths Example, page 549](#)

Configuring Event-Based VPN Import Processing for BGP Paths Example

In this example configuration, a VRF (vrf-A) is configured and VRF forwarding is applied to Fast Ethernet interface 1/1. In address family mode the import path selection is set to all and the number of import paths is set to 3. Two BGP neighbors are configured under the IPv4 address family and activated under the VPNv4 address family.

```

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

```

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" module.
- If you want to configure BGP neighbor session options, see the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some advanced BGP features, see the "Configuring Advanced BGP Features" module.

Additional References

The following sections provide references related to the BGP Event-Based VPN Import feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	"Cisco BGP Overview" module of the <i>Cisco IOS IP Routing: BGP Configuration Guide</i> .
Conceptual and configuration details for basic BGP tasks.	"Configuring a Basic BGP Network" module of the <i>Cisco IOS IP Routing Protocols Configuration Guide</i> .
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup
<i>Cisco IOS Master Command List</i>	http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

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 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
<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/techsupport

Feature Information for BGP Event-Based VPN Import

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

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

Table 36 **Feature Information for BGP Event-Based VPN Import**

Feature Name	Releases	Feature Information
BGP Event-Based VPN Import	12.2(33)SRE 15.0(1)M 15.0(1)S Cisco IOS XE 3.1.0SG, 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>

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: 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)

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



BGP Best External

The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. The BGP Best External feature advertises the most preferred route among those received from external neighbors as a backup route. This feature is beneficial in active-backup topologies, where service providers use routing policies that cause a border router to choose a path received over an internal BGP (iBGP) session (of another border router) as the best path for a prefix even if it has an external BGP (eBGP) learned path. This active-backup topology defines one exit or egress point for the prefix in the autonomous system and uses the other points as backups if the primary link or eBGP peering is unavailable. The policy causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because it does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path.

- [Finding Feature Information, page 553](#)
- [Contents, page 553](#)
- [Prerequisites for BGP Best External, page 554](#)
- [Restrictions for BGP Best External, page 554](#)
- [Information About BGP Best External, page 554](#)
- [How to Configure BGP Best External, page 556](#)
- [Configuration Examples for BGP Best External, page 561](#)
- [Additional References, page 562](#)
- [Feature Information for BGP Best External, page 563](#)

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, page 563](#).

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.

Contents

Prerequisites for BGP Best External

- The Bidirectional Forwarding Detection (BFD) protocol must be enabled to quickly detect link failures.
- Ensure that the BGP and the Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- The backup path must have a unique next hop that is not the same as the next hop of the best path.
- BGP must support lossless switchover between operational paths.

Restrictions for BGP Best External

- The BGP Best External feature will not install a backup path if BGP Multipath is installed and a multipath exists in the BGP table. One of the multipaths automatically acts as a backup for the other paths.
- The BGP Best External feature is not supported with the following features:
 - MPLS VPN Carrier Supporting Carrier
 - MPLS VPN Inter-Autonomous Systems, option B
 - MPLS VPN Per Virtual Routing and Forwarding (VRF) Label
- The BGP Best External feature cannot be configured with Multicast or L2VPN VRF address families.
- The BGP Best External feature cannot be configured on route reflectors.
- The BGP Best External feature does not support NSF/SSO. However, ISSU is supported if both Route Processors have the BGP Best External feature configured.
- The BGP Best External feature can only be configured on VPNv4, VPNv6, IPv4 VRF, and IPv6 VRF address families.
- When you configure the BGP Best External feature using the **bgp advertise-best-external** command, you need not enable the BGP PIC feature with the **bgp additional-paths install** command. The BGP PIC feature is automatically enabled by the BGP Best External feature.
- When you configure the BGP Best External feature, it will override the functionality of the [MPLS VPN--BGP Local Convergence](#) feature. However, you do not have to remove the **protection local-prefixes** command from the configuration.

Information About BGP Best External

- [BGP Best External Overview](#), page 554
- [What the Best External Route Means](#), page 555
- [How the BGP Best External Feature Works](#), page 555
- [Configuration Modes for Enabling BGP Best External](#), page 556

BGP Best External Overview

Service providers use routing policies that cause a border router to choose a path received over an iBGP session (of another border router) as the best path for a prefix even if it has an eBGP learned path. This practice is popularly known as active-backup topology and is done to define one exit or egress point for the

prefix in the autonomous system and to use the other points as backups if the primary link or eBGP peering is unavailable.

The policy, though beneficial, causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because the border router does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path. The best external behavior causes the BGP selection process to select two paths to every destination:

- The best path is selected from the complete set of routes known to that destination.
- The best external path is selected from the set of routes received from its external peers.

BGP advertises the best path to external peers. Instead of withdrawing the best path from its internal peers when it selects an iBGP path as the best path, BGP advertises the best external path to the internal peers.

The BGP Best External feature is an essential component of the Prefix-Independent Convergence (PIC) edge for both Internet access and MPLS VPN scenarios and makes alternate paths available in the network in the active-backup topology.

What the Best External Route Means

The BGP Best External feature uses a "best external route" as a backup path, which, according to *draft-marques-idr-best-external*, is the most preferred route among those received from external neighbors. The most preferred route from external neighbors can be the following:

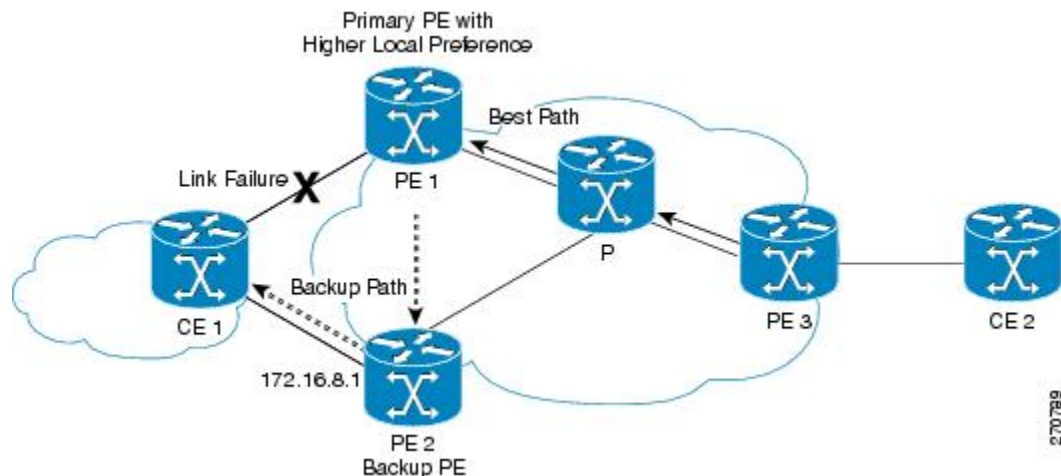
- Two routers in different clusters that have an iBGP session between them.
- Two routers in different autonomous systems of a confederation that have an eBGP session between them.

The best external route might be different from the best route installed in the routing information base (RIB). The best route could be an internal route. By allowing the best external route to be advertised and stored, in addition to the best route, networks gain faster restoration of connectivity by providing additional paths that may be used if the primary path fails.

How the BGP Best External Feature Works

The BGP Best External feature is based on Internet Engineering Task Force (IETF) draft-marques-idr-best-external.txt. The BGP Best External feature advertises a best external route to its internal peers as a backup route. The backup route is stored in the RIB and Cisco Express Forwarding. If the primary path fails, the BGP PIC functionality enables the best external path to take over, enabling faster restoration of connectivity.

Figure 47 MPLS VPN: Best External at the Edge of MPLS VPN



[How the BGP Best External Feature Works, page 555](#) shows an MPLS VPN using the BGP Best External feature. The network includes the following components:

- eBGP sessions exist between the provider edge (PE) and customer edge (CE) routers.
- PE1 is the primary router and has a higher local preference setting.
- Traffic from CE2 uses PE1 to reach router CE1.
- PE1 has two paths to reach CE1.
- CE1 is dual-homed with PE1 and PE2.
- PE1 is the primary path and PE2 is the backup path.

In [How the BGP Best External Feature Works, page 555](#), traffic in the MPLS cloud flows through PE1 to reach CE1. Therefore, PE2 uses PE1 as the best path and PE2 as the backup path.

PE1 and PE2 are configured with the BGP Best External feature. BGP computes both the best path (the PE1-CE1 link) and a backup path (PE2) and installs both paths into the RIB and Cisco Express Forwarding. The best external path (PE2) is advertised to the peer routers, in addition to the best path.

When Cisco Express Forwarding detects a link failure on the PE1-CE1 link, Cisco Express Forwarding immediately switches to the backup path PE2. Traffic is quickly rerouted due to local Fast Convergence in Cisco Express Forwarding using the backup path. Thus, traffic loss is minimized and fast convergence is achieved.

Configuration Modes for Enabling BGP Best External

You can enable the BGP Best External feature in different modes, each of which protects VRFs in its own way:

- If you issue the **bgp advertise-best-external** command in VPNv4 address family configuration mode, it applies to all IPv4 VRFs. If you issue the command in this mode, you need not issue it for specific VRFs.
- If you issue the **bgp advertise-best-external** command in IPv4 address family configuration mode, it applies only that VRF.

How to Configure BGP Best External

- [Enabling the BGP Best External Feature, page 556](#)
- [Verifying the BGP Best External Feature, page 559](#)

Enabling the BGP Best External Feature

Perform the following task to enable the BGP Best External feature. This task shows how to configure the BGP Best External feature in either IPv4 or VPNv4 address family. In VPNv4 address family configuration mode, the BGP Best External feature applies to all IPv4 VRFs; you do not have to configure it for specific VRFs. If you issue the **bgp advertise-best-external** command in IPv4 VRF address family configuration mode, the BGP Best External feature applies only that VRF.

- Configure the MPLS VPN and verify that it is working properly before configuring the BGP Best External feature. See "Configuring MPLS Layer 3 VPNs" for more information.
- Configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or configure separate route-target policies for IPv4 and IPv6 VPNs. For

information about configuring multiprotocol VRFs, see "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs".

- Ensure that the CE router is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
 - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
 - or
 - **address-family vpnv4** [**unicast**]
5. **bgp advertise-best-external**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **neighbor** *ip-address* **fall-over** [**b fd**|**route-map** *map-name*]
9. **end**

DETAILED STEPS

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
<p>Step 4 Do one of the following:</p> <ul style="list-style-type: none"> • address-family ipv4 [unicast vrf <i>vrf-name</i>] • or • address-family vpv4 [unicast] <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre> <p>Example:</p> <pre>Router(config-router)# address-family vpv4</pre>	<p>Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.</p> <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.
<p>Step 5 bgp advertise-best-external</p> <p>Example:</p> <pre>Router(config-router-af)# bgp advertise- best-external</pre>	<p>Calculates and uses an external backup path and installs it into the RIB and Cisco Express Forwarding.</p>
<p>Step 6 neighbor ip-address 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.
<p>Step 7 neighbor ip-address 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>

Command or Action	Purpose
<p>Step 8 <code>neighbor ip-address fall-over [b fd route-map map-name]</code></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.
<p>Step 9 <code>end</code></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

- enable
- show vrf detail
- 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]
- show bgp vpnv4 unicast vrf vrf-name ip-address
- show ip route vrf vrf-name repair-paths ip-address
- 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-regexp] [regexp] [summary] [labels]

Use this command to verify that the best external route is advertised. In the command output, the code b indicates a backup path and the code x designates the best external path.

Example:

```
Router# show ip bgp vpnv4 all
BGP table version is 1104964, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, multipath,
b backup-path, x best-external
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 11:12 (default for vrf blue)
*>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, page 561](#)

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 Commands List, All Releases
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Routing: BGP Command Reference</i>
Basic MPLS VPNs	"Configuring MPLS Layer 3 VPNs"
Multiprotocol VRFs	"MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs"
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	BGP Best External, Advertisement of the best external route to iBGP

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 2547	<i>BGP/MPLS VPNs</i>
RFC 1771	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 Best External

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

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

Table 37 **Feature Information for BGP Best External**

Feature Name	Releases	Feature Information
BGP Best External	12.2(33)SRE 15.2(3)T	<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>In Cisco IOS Release 12.2(33)SRE, this feature was introduced.</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, show ip route vrf.</p>

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: 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)

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



BGP 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 565](#)
- [Contents, page 565](#)
- [Prerequisites for BGP PIC, page 566](#)
- [Restrictions for BGP PIC, page 566](#)
- [Information About BGP PIC, page 566](#)
- [How to Configure BGP PIC, page 574](#)
- [Configuration Examples for BGP PIC, page 577](#)
- [Additional References, page 580](#)
- [Feature Information for BGP PIC, page 582](#)

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 PIC, page 582](#).

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.

Contents

Prerequisites for BGP PIC

- Ensure that the Border Gateway Protocol (BGP) and the IP or Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- Ensure that the backup/alternate path has a unique next hop that is not the same as the next hop of the best path.
- Enable the Bidirectional Forwarding Detection (BFD) protocol to quickly detect link failures of directly connected neighbors.

Restrictions for BGP PIC

- With BGP Multipath, the BGP Prefix-Independent Convergence (PIC) feature is already supported.
- In MPLS VPNs, the BGP PIC feature is not supported with MPLS VPN Inter-Autonomous Systems Option B.
- The BGP PIC feature supports prefixes only for IPv4, IPv6, VPNv4, and VPNv6 address families.
- The BGP PIC feature cannot be configured with Multicast or L2VPN Virtual Routing and Forwarding (VRF) address families.
- If the route reflector is only in the control plane, then you do not need BGP PIC, because BGP PIC addresses data plane convergence.
- When two PE routers become each other's backup/alternate path to a CE router, traffic might loop if the CE router fails. Neither router will reach the CE router, and traffic will continue to be forwarded between the PE routers until the time-to-live (TTL) timer expires.
- The BGP PIC feature does not support Nonstop Forwarding with Stateful Switchover (NSF/SSO). However, ISSU is supported if both Route Processors have the BGP PIC feature configured.
- The BGP PIC feature solves the traffic forwarding only for a single network failure at both the edge and the core.
- The BGP PIC feature does not work with the BGP Best External feature. If you try to configure the BGP PIC feature after configuring the BGP Best External feature, you receive an error.

Information About BGP PIC

- [Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature, page 567](#)
- [How BGP Converges Under Normal Circumstances, page 567](#)
- [How BGP PIC Improves Convergence, page 567](#)
- [How a Failure Is Detected, page 569](#)
- [How BGP PIC Achieves Subsecond Convergence, page 569](#)
- [How BGP PIC Improves Upon the Functionality of MPLS VPN--BGP Local Convergence, page 570](#)
- [Configuration Modes for Enabling BGP PIC, page 570](#)
- [BGP PIC Scenarios, page 570](#)
- [Cisco Express Forwarding Recursion, page 574](#)

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, page 568](#)

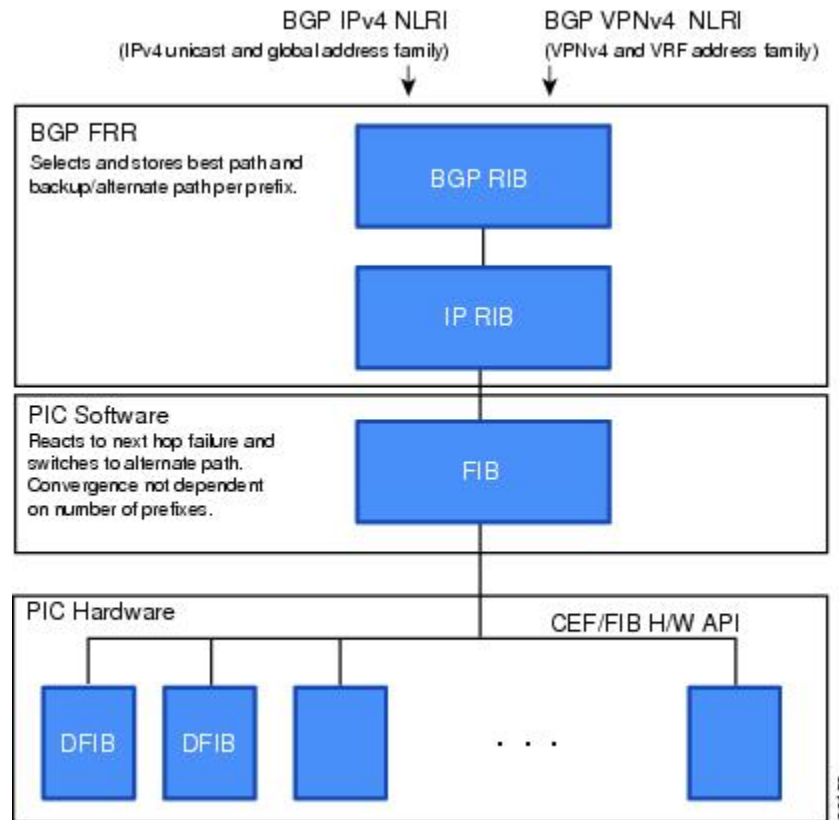
BGP Fast Reroute's Role in the BGP PIC Feature

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 48 *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

The BGP PIC feature works at the Cisco Express Forwarding level, and Cisco Express Forwarding can be processed in both hardware line cards and in the software.

- For platforms that support Cisco Express Forwarding processing in the line cards, the BGP PIC feature can converge in subseconds. The Cisco 7600 router and Cisco 10000 router supports Cisco Express Forwarding processing in the line cards and in the software, and thus can attain subsecond convergence.
- For platforms that do not use Cisco Express Forwarding in hardware line cards, Cisco Express Forwarding is achieved in the software. The BGP PIC feature will work with the Cisco Express

Forwarding through the software and achieve convergence within seconds. The Cisco 7200 router supports Cisco Express Forwarding in the software and thus can achieve convergence in seconds rather than milliseconds.

How BGP PIC Improves Upon the Functionality of MPLS VPN--BGP Local Convergence

The BGP PIC feature is an enhancement to the [MPLS VPN--BGP Local Convergence](#) feature, which provides a failover mechanism that recalculates the best path and installs the new path in forwarding after a link failure. The feature maintains the local label for 5 minutes to ensure that the traffic uses the backup/alternate path, thus minimizing traffic loss.

The BGP PIC feature improves the LoC time to under a second by calculating a backup/alternate path in advance. When a link failure occurs, the traffic is sent to the backup/alternate path.

When you configure the BGP PIC feature, it will override the functionality of the [MPLS VPN--BGP Local Convergence](#) feature. You do not have to remove the **protection local-prefixes** command from the configuration.

Configuration Modes for Enabling BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

BGP PIC Scenarios

The following scenarios explain how you can configure the BGP PIC functionality to achieve fast convergence:

- [IP PE-CE Link and Node Protection on the CE Side \(Dual PEs\)](#), page 570
- [IP PE-CE Link and Node Protection on the CE Side \(Dual CEs and Dual PE Primary and Backup Nodes\)](#), page 571
- [IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path](#), page 572
- [IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path](#), page 573

IP PE-CE Link and Node Protection on the CE Side (Dual PEs)

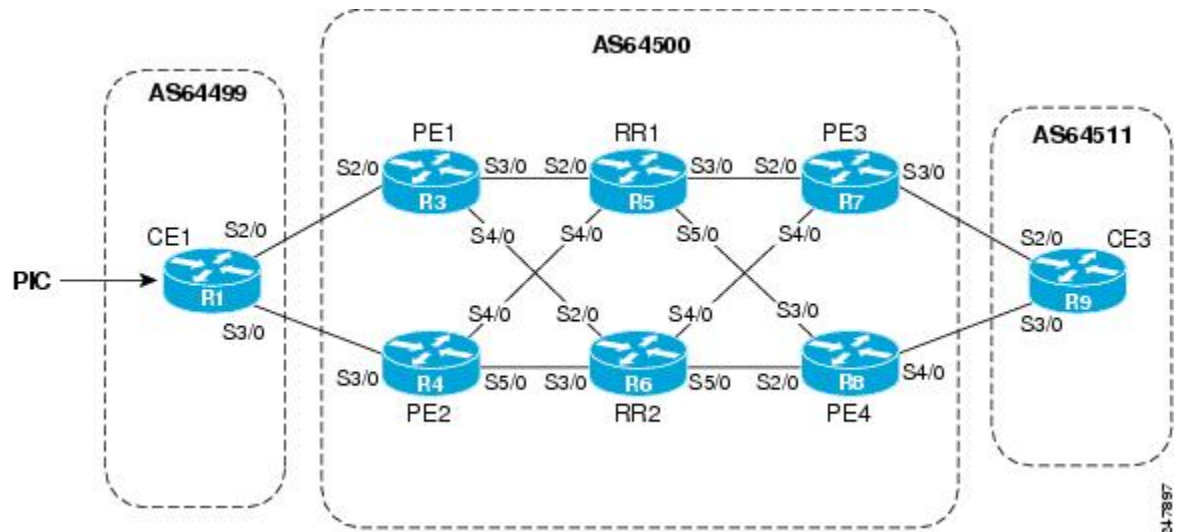
The figure below shows a network that uses the BGP PIC feature. The network includes the following components:

- 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 49 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.

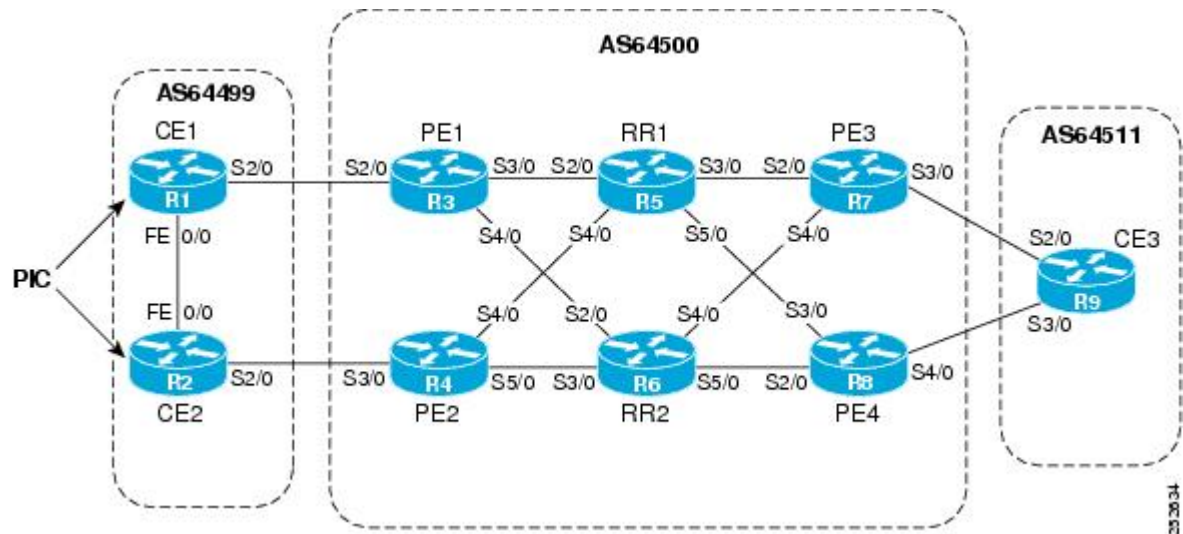
There should not be any policies set on CE1 and CE2 for the eBGP peers PE1 and PE2. Both CE routers must point to the eBGP route as next hop. On CE1, the next hop to reach CE3 is through PE1, so PE1 is the best path to reach CE3. On CE2, the best path to reach CE3 is PE2. CE2 advertises itself as the 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 50 Using BGP PIC in a Dual CE, Dual PE Network



IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path

[IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path, page 572](#) shows a network that uses the BGP PIC feature on CE1 and CE2. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach 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.

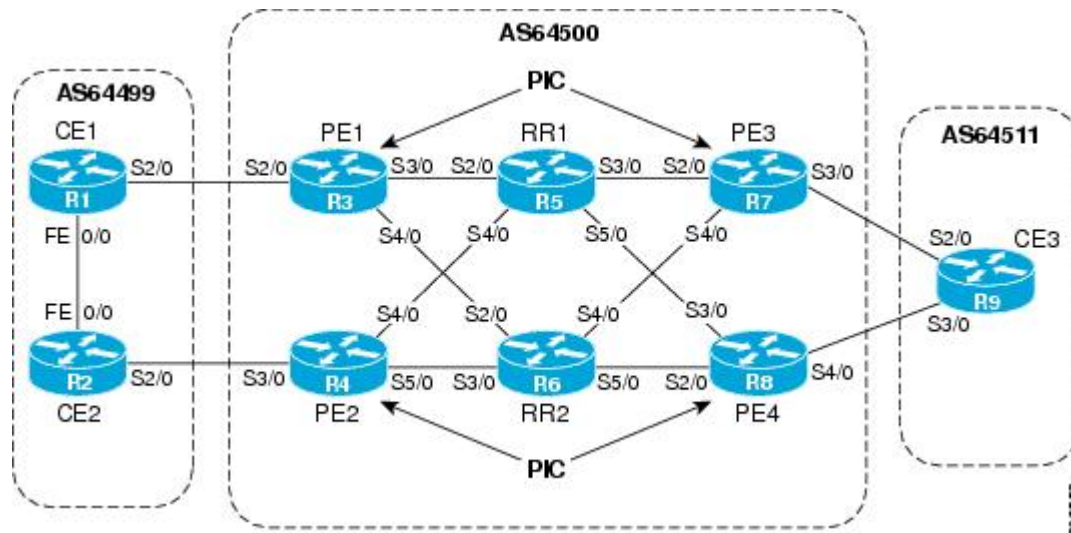
Thus, with BGP PIC enabled on PE3 and PE4, Cisco Express Forwarding detects the link failure and does in-place modification of the forwarding object to the backup/alternate node PE4 that already exists in Cisco

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

[IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path](#), page 573 shows a network that uses the BGP PIC feature on all the PE routers in an MPLS network.

Figure 51 Enabling BGP PIC on All PEs Routers in the MPLS Network



The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach 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.

To disable or enable Cisco Express Forwarding recursion for BGP directly connected next hops, you can issue the **disable-connected-check** command.

How to Configure BGP PIC

- [Configuring BGP PIC, page 575](#)

Configuring BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

For a full configuration example that includes configuring multiprotocol VRFs and shows output to verify that the feature is enabled, see the [Example: Configuring BGP PIC, page 578](#).

- If you are implementing the BGP PIC feature in an MPLS VPN, ensure that the network is working properly before configuring the BGP PIC feature. See [Configuring MPLS Layer 3 VPNs](#) for more information.
- If you are implementing the BGP PIC feature in an MPLS VPN, configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see [MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs](#).
- Ensure that the CE router is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
 - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
 -
 - or
 - **address-family vpnv4** [**unicast**]
5. **bgp additional-paths install**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **bgp recursion host**
9. **neighbor** *ip-address* **fall-over** [**bfd** | **route-map** *map-name*]
10. **end**

DETAILED STEPS

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

Command or Action	Purpose
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 router bgp <i>autonomous-system-number</i></p> <p>Example:</p> <pre>Router(config)# router bgp 40000</pre>	<p>Enters router configuration mode for the specified routing process.</p>
<p>Step 4 Do one of the following:</p> <ul style="list-style-type: none"> • address-family ipv4 [unicast vrf <i>vrf-name</i>] • • or • address-family vpnv4 [unicast] <p>Example:</p> <pre>Router(config-router)# address-family ipv4 unicast</pre> <p>Example:</p> <p>Example:</p> <pre>Router(config-router)# address-family vpn4</pre>	<p>Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.</p> <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 with subsequent IPv4 address family configuration mode commands.
<p>Step 5 bgp additional-paths install</p> <p>Example:</p> <pre>Router(config-router-af)# bgp additional- paths install</pre>	<p>Calculates a backup/alternate path and installs it into the RIB and Cisco Express Forwarding.</p>

	Command or Action	Purpose
Step 6	<p>neighbor ip-address remote-as autonomous-system-number</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 ip-address 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>bgp recursion host</p> <p>Example:</p> <pre>Router(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>Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd</pre>	<p>Enables BFD protocol support to detect when a neighbor has gone away, which can occur within a subsecond.</p>
Step 10	<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>

Configuration Examples for BGP PIC

- [Example: Configuring BGP PIC, page 578](#)
- [Example: Displaying Backup Alternate Paths for BGP PIC, page 579](#)

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
!
interface Ethernet1/0
 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 redistribute static
 redistribute connected
 neighbor 10.6.6.6 remote-as 3
 neighbor 10.6.6.6 update-source Loopback0
 neighbor 10.7.7.7 remote-as 3
 neighbor 10.7.7.7 update-source Loopback0
 no auto-summary
!
address-family vpnv4
 bgp additional-paths install
 neighbor 10.6.6.6 activate
 neighbor 10.6.6.6 send-community both
 neighbor 10.7.7.7 activate
 neighbor 10.7.7.7 send-community both
 exit-address-family
!
address-family ipv4 vrf blue
 import path selection all
 import path limit 10
 no synchronization
 neighbor 10.11.11.11 remote-as 1
 neighbor 10.11.11.11 activate
 exit-address-family
!
address-family ipv4 vrf green
 import path selection all
 import path limit 10
 no synchronization
 neighbor 10.13.13.13 remote-as 1
 neighbor 10.13.13.13 activate
 exit-address-family
```

The following **show vrf detail** command output shows that the BGP PIC feature is enabled:

```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
  Interfaces:
    Se4/0
  Address family ipv4 (Table ID = 1 (0x1)):
    Export VPN route-target communities
      RT:100:1          RT:200:1          RT:300:1
```



```

RT:400:1
Import VPN route-target communities
RT:100:1          RT:200:1          RT:300:1
RT:400:1
No import route-map
No export route-map
VRF label distribution protocol: not configured
VRF label allocation mode: per-prefix
Prefix protection with additional path enabled
Address family ipv6 not active.

```

Example: Displaying Backup Alternate Paths for BGP PIC

The command output in the following example shows that the VRFs in VRF blue have backup/alternate paths:

```

Router# show ip bgp vpnv4 vrf blue 10.0.0.0
BGP routing table entry for 10:12:12.0.0.0/24, version 88
Paths: (4 available, best #1, table blue)
  Additional-path
  Advertised to update-groups:
    6
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      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:

```

Router# show ip bgp vpnv4 vrf green 12.0.0.0
BGP routing table entry for 12:23:12.0.0.0/24, version 87
Paths: (4 available, best #4, table green)
  Additional-path
  Advertised to update-groups:
    5
  1, imported path from 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:

```
Router# show ip bgp 10.0.0.0 255.255.0.0
BGP routing table entry for 10.0.0.0/16, version 123
Paths: (4 available, best #3, table default)
  Additional-path
  Advertised to update-groups:
    2          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:

```
Router# show ip route repair-paths 10.0.0.0 255.255.0.0
Routing entry for 10.0.0.0/16
  Known via "bgp 10", distance 200, metric 0, type internal
  Last update from 10.0.101.2 00:00:56 ago
  Routing Descriptor Blocks:
  * 10.0.101.2, from 10.0.101.2, 00:00:56 ago
    Route metric is 0, traffic share count is 1
    AS Hops 0
    MPLS label: none
  [RPR]10.0.101.1, from 10.0.101.1, 00:00:56 ago
    Route metric is 0, traffic share count is 1
    AS Hops 0
    MPLS label: none
```

The command output in the following example shows the Cisco Express Forwarding/forwarding information base entries for the backup and alternate paths:

```
Router# show ip cef 10.0.0.0 255.255.0.0 detail
10.0.0.0/16, epoch 0, flags rib only nolabel, rib defined all labels
  recursive via 10.0.101.2
    attached to GigabitEthernet0/2
  recursive via 10.0.101.1, repair
    attached to GigabitEthernet0/2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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
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
BGP routing	BGP Feature Roadmap

Standards

Standard	Title
draft-walton-bgp-add-paths-04.txt	<i>Advertisement of Multiple Paths in BGP</i>

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 2547	<i>BGP/MPLS VPNs</i>
RFC 1771	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 PIC

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

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

Table 38 **Feature Information for BGP PIC**

Feature Name	Releases	Feature Information
BGP PIC Edge for IP and MPLS-VPN	12.2(33)SRE 12.2(33)XNE 15.0(1)S 15.2(3)T	<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>In 12.2(33)SRE, this feature was introduced on the Cisco 7200 and Cisco 7600 routers.</p> <p>In 12.2(33)XNE, support was added for the Cisco 10000 router.</p> <p>The following sections provide information about this feature:</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, show vrf.</p>

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: 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)

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.

