



- [Table of Contents](#)
- [Index](#)

## **Cisco® BGP-4 Command and Configuration Handbook (CCIE Professional Development)**

By William R. Parkhurst CCIE #2969 Ph.D.,

Publisher: Cisco Press

Pub Date: May 08, 2001

ISBN: 1-58705-017-X

Pages: 250

Slots: 2

As the means by which Internet destinations are communicated between subscribers and service providers, BGP-4 provides a critical function for Internet operations. Whether associated with Internet service providers or enterprise networks that depend heavily on the Internet, a background in BGP-4 is essential. Moving beyond background information, BGP contains a great number of commands, and practical information on those commands is difficult to come by.

Cisco BGP-4 Command and Configuration Handbook is an exhaustive practical reference to the commands contained within BGP-4. For each command/subcommand, author Bill Parkhurst explains the intended use or function and how to properly configure it. Then he presents scenarios to demonstrate every facet of the command and its use, along with appropriate show and debug commands. Through the discussion of functionality and the scenario-based configuration examples, Cisco BGP-4 Command and Configuration Handbook will help you gain a thorough understanding of the practical side of BGP-4.



- [Table of Contents](#)
- [Index](#)

## Cisco® BGP-4 Command and Configuration Handbook (CCIE Professional Development)

By William R. Parkhurst CCIE #2969 Ph.D.,

Publisher: Cisco Press

Pub Date: May 08, 2001

ISBN: 1-58705-017-X

Pages: 250

Slots: 2

[Copyright](#)

[About the Author](#)

[About the Technical Reviewers](#)

[Acknowledgments](#)

[Introduction](#)

[Icons Used in This Book](#)

[Command Syntax Conventions](#)

[Chapter 1. Route Aggregation](#)

[Section 1-1. aggregate-address](#) address mask

[Section 1-2. aggregate-address](#) address mask [as-set](#)

[Section 1-3. aggregate-address](#) address mask [as-set advertise-map](#) route-map-name

[Section 1-4. aggregate-address](#) address mask [attribute-map](#) route-map-name

[Section 1-5. aggregate-address](#) address mask [route-map](#) route-map-name

[Section 1-6. aggregate-address](#) address mask [summary-only](#)

[Section 1-7. aggregate-address](#) address mask [suppress-map](#) route-map-name

[Chapter 2. Auto-Summary](#)

[Section 2-1. auto-summary](#)

[Chapter 3. BGP-Specific Commands](#)

[Section 3-1. bgp always-compare-med](#)

[Section 3-2. bgp bestpath as-path ignore](#)

[Section 3-3. bgp bestpath med confed](#)

[Section 3-4. bgp bestpath med missing-as-worst](#)

[Section 3-5. bgp client-to-client reflection](#)

[Section 3-6. bgp cluster-id 32-bit\\_id](#)  
[Section 3-7. bgp confederation identifier AS-number](#)  
[Section 3-8. bgp confederation peers 1\\_or\\_more\\_AS-numbers](#)  
[Section 3-9. bgp dampening](#)  
[Section 3-10. bgp dampening half-life](#)  
[Section 3-11. bgp dampening half-life reuse suppress max-suppress-time](#)  
[Section 3-12. bgp dampening route-map route-map-name](#)  
[Section 3-13. bgp default local-preference local-preference](#)  
[Section 3-14. bgp deterministic-med](#)  
[Section 3-15. bgp fast-external-fallover](#)  
[Section 3-16. bgp log-neighbor-changes](#)  
[Section 3-17. bgp router-id ip-address](#)

#### [Chapter 4. Default Information](#)

[Section 4-1. default-information originate](#)  
[Section 4-2. default-metric metric](#)

#### [Chapter 5. BGP Administrative Distance](#)

[Section 5-1. distance admin-distance ip-source-address ip-address-mask](#)  
[Section 5-2. distance admin-distance ip-source-address ip-address-mask ip-access-list-number](#)  
[Section 5-3. distance bgp external internal local](#)

#### [Chapter 6. BGP Route Filtering](#)

[Section 6-1. distribute-list](#)

#### [Chapter 7. BGP Maximum Paths](#)

[Section 7-1. maximum-paths number-of-paths](#)

#### [Chapter 8. Neighbor Configuration](#)

[Section 8-1. neighbor { ip-address | peer-group-name } advertise-map route-map-name1 non-exist-map route-map-name2](#)  
[Section 8-2. neighbor { ip-address | peer-group-name } advertisement-interval seconds](#)  
[Section 8-3. neighbor { ip-address | peer-group-name } default-originate](#)  
[Section 8-4. neighbor { ip-address | peer-group-name } default-originate route-map route-map-name](#)  
[Section 8-5. neighbor { ip-address | peer-group-name } description text](#)  
[Section 8-6. neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name prefix-list-name } in](#)  
[Section 8-7. neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name | prefix-list-name } out](#)  
[Section 8-8. neighbor { ip-address | peer-group-name } ebgp-multihop](#)  
[Section 8-9. neighbor { ip-address | peer-group-name } ebgp-multihop maximum-hop-count](#)  
[Section 8-10. neighbor { ip-address | peer-group-name } filter-list as-path-list-number in](#)  
[Section 8-11. neighbor { ip-address | peer-group-name } filter-list as-path-list-number out](#)  
[Section 8-12. neighbor { ip-address | peer-group-name } filter-list as-path-list-number weight weight](#)

[Section 8-13. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit](#)  
[Section 8-14. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit warning-only](#)  
[Section 8-15. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit threshold-value](#)  
[Section 8-16. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit threshold-value warning-only](#)  
[Section 8-17. neighbor { ip-address | peer-group-name } next-hop-self](#)  
[Section 8-18. neighbor { ip-address | peer-group-name } password password](#)  
[Section 8-19. neighbor peer-group-name peer-group](#)  
[Section 8-20. neighbor ip-address peer-group peer-group-name](#)  
[Section 8-21. : neighbor { ip-address | peer-group-name } prefix-list prefix-list-name in](#)  
[Section 8-22. neighbor { ip-address | peer-group-name } prefix-list prefix-list-name out](#)  
[Section 8-23. neighbor { ip-address | peer-group-name } remote-as number](#)  
[Section 8-24. neighbor { ip-address | peer-group-name } remove-private-as](#)  
[Section 8-25. neighbor { ip-address | peer-group-name } route-map route-map-name in](#)  
[Section 8-26. neighbor { ip-address | peer-group-name } route-map route-map-name out](#)  
[Section 8-27. neighbor { ip-address | peer-group-name } route-reflector-client](#)  
[Section 8-28. neighbor { ip-address | peer-group-name } send-community](#)  
[Section 8-29. neighbor { ip-address | peer-group-name } shutdown](#)  
[Section 8-30. neighbor { ip-address | peer-group-name } soft-reconfiguration inbound](#)  
[Section 8-31. neighbor { ip-address | peer-group-name } timers keepalive holdtime](#)  
[Section 8-32. neighbor { ip-address | peer-group-name } unsuppress-map route-map-name](#)  
[Section 8-33. neighbor { ip-address | peer-group-name } update-source interface-name](#)  
[Section 8-34. neighbor { ip-address | peer-group-name } version version-number](#)  
[Section 8-35. neighbor { ip-address | peer-group-name } weight default-weight](#)

## [Chapter 9. Route Advertisement](#)

[Section 9-1. network ip-address](#)  
[Section 9-2. network ip-address mask network-mask](#)  
[Section 9-3. network ip-address backdoor](#)  
[Section 9-4. network ip-address mask network-mask backdoor](#)  
[Section 9-5. network ip-address route-map route-map-name](#)  
[Section 9-6. network ip-address mask network-mask route-map route-map-name](#)  
[Section 9-7. network ip-address weight weight](#)  
[Section 9-8. network ip-address mask network-mask weight weight](#)

## [Chapter 10. Route Redistribution](#)

[Section 10-1. redistribute protocol](#)  
[Section 10-2. redistribute protocol metric metric](#)  
[Section 10-3. redistribute protocol route-map route-map-name](#)  
[Section 10-4. redistribute protocol route-map route-map-name metric metric](#)  
[Section 10-5. redistribute protocol weight weight](#)

## [Chapter 11. Address Summarization](#)

[Section 11-1. summary-address](#)

## [Chapter 12. Synchronization](#)

### [Section 12-1. synchronization](#)

## [Chapter 13. BGP Timers](#)

### [Section 13-1. timers bgp keepalive holdtime](#)

## [Chapter 14. BGP show Commands](#)

### [Section 14-1. show ip bgp](#)

### [Section 14-2. show ip bgp | begin line](#)

### [Section 14-3. show ip bgp | exclude line](#)

### [Section 14-4. show ip bgp | include line](#)

### [Section 14-5. show ip bgp prefix](#)

### [Section 14-6. show ip bgp prefix mask](#)

### [Section 14-7. show ip bgp prefix/mask-length](#)

### [Section 14-8. show ip bgp prefix mask longer-prefixes](#)

### [Section 14-9. show ip bgp prefix/mask-length longer-prefixes](#)

### [Section 14-10. show ip bgp prefix | begin line](#)

### [Section 14-11. show ip bgp prefix | exclude line](#)

### [Section 14-12. show ip bgp prefix | include line](#)

### [Section 14-13. show ip bgp prefix mask | begin line](#)

### [Section 14-14. show ip bgp prefix mask | exclude line](#)

### [Section 14-15. show ip bgp prefix mask | include line](#)

### [Section 14-16. show ip bgp prefix/mask-length | begin line](#)

### [Section 14-17. show ip bgp prefix/mask-length | exclude line](#)

### [Section 14-18. show ip bgp prefix/mask-length | include line](#)

### [Section 14-19. show ip bgp prefix mask longer-prefixes | begin line](#)

### [Section 14-20. show ip bgp prefix mask longer-prefixes | exclude line](#)

### [Section 14-21. show ip bgp prefix mask longer-prefixes | include line](#)

### [Section 14-22. show ip bgp prefix/mask-length longer-prefixes | begin line](#)

### [Section 14-23. show ip bgp prefix/mask-length longer-prefixes | exclude line](#)

### [Section 14-24. show ip bgp prefix/mask-length longer-prefixes | include line](#)

### [Section 14-25. show ip bgp cidr-only](#)

### [Section 14-26. show ip bgp cidr-only | begin line](#)

### [Section 14-27. show ip bgp cidr-only | exclude line](#)

### [Section 14-28. show ip bgp cidr-only | include line](#)

### [Section 14-29. show ip bgp community community-number\(s\)](#)

### [Section 14-30. show ip bgp community community-number\(s\) exact-match](#)

### [Section 14-31. show ip bgp community community-number\(s\) | begin line](#)

### [Section 14-32. show ip bgp community community-number\(s\) | exclude line](#)

### [Section 14-33. show ip bgp community community-number\(s\) | include line](#)

### [Section 14-34. show ip bgp community community-number\(s\) | begin line exact-match](#)

### [Section 14-35. show ip bgp community community-number\(s\) | exclude line exact-match](#)

### [Section 14-36. show ip bgp community community-number\(s\) | include line exact-match](#)

### [Section 14-37. show ip bgp community-list community-list-number](#)

### [Section 14-38. show ip bgp community-list community-list-number | begin line](#)

### [Section 14-39. show ip bgp community-list community-list-number | exclude line](#)

### [Section 14-40. show ip bgp community-list community-list-number | include line](#)

[Section 14-41. show ip bgp community-list community-list-number exact-match](#)  
[Section 14-42. show ip bgp community-list community-list-number exact-match begin](#)  
line  
[Section 14-43. show ip bgp community-list community-list-number exact-match exclude](#)  
line  
[Section 14-44. show ip bgp community-list community-list-number exact-match include](#)  
line  
[Section 14-45. show ip bgp dampened-paths](#)  
[Section 14-46. show ip bgp dampened-paths begin](#) line  
[Section 14-47. show ip bgp dampened-paths exclude](#) line  
[Section 14-48. show ip bgp dampened-paths include](#) line  
[Section 14-49. show ip bgp filter-list as-path-access-list](#)  
[Section 14-50. show ip bgp filter-list as-path-access-list begin](#) line  
[Section 14-51. show ip bgp filter-list as-path-access-list exclude](#) line  
[Section 14-52. show ip bgp filter-list as-path-access-list include](#) line  
[Section 14-53. show ip bgp flap-statistics](#)  
[Section 14-54. show ip bgp flap-statistics prefix/mask-length](#)  
[Section 14-55. show ip bgp flap-statistics prefix/mask-length longer-prefixes](#)  
[Section 14-56. show ip bgp flap-statistics prefix/mask-length longer-prefixes begin](#)  
regular-expression  
[Section 14-57. show ip bgp flap-statistics prefix/mask-length longer-prefixes exclude](#)  
regular-expression  
[Section 14-58. show ip bgp flap-statistics prefix/mask-length longer-prefixes include](#)  
regular-expression  
[Section 14-59. show ip bgp flap-statistics prefix/mask-length begin](#) regular-expression  
[Section 14-60. show ip bgp flap-statistics prefix/mask-length exclude](#) regular-expression  
[Section 14-61. show ip bgp flap-statistics prefix/mask-length include](#) regular-expression  
[Section 14-62. show ip bgp flap-statistics prefix](#)  
[Section 14-63. show ip bgp flap-statistics prefix begin](#) regular-expression  
[Section 14-64. show ip bgp flap-statistics prefix exclude](#) regular-expression  
[Section 14-65. show ip bgp flap-statistics prefix include](#) regular-expression  
[Section 14-66. show ip bgp flap-statistics prefix mask](#)  
[Section 14-67. show ip bgp flap-statistics prefix mask begin](#) regular-expression  
[Section 14-68. show ip bgp flap-statistics prefix mask exclude](#) regular-expression  
[Section 14-69. show ip bgp flap-statistics prefix mask include](#) regular-expression  
[Section 14-70. show ip bgp flap-statistics prefix mask longer-prefixes](#)  
[Section 14-71. show ip bgp flap-statistics prefix mask longer-prefixes begin](#)  
regular-expression  
[Section 14-72. show ip bgp flap-statistics prefix mask longer-prefixes exclude](#)  
regular-expression  
[Section 14-73. show ip bgp flap-statistics prefix mask longer-prefixes include](#)  
regular-expression  
[Section 14-74. show ip bgp flap-statistics filter-list list-number](#)  
[Section 14-75. show ip bgp flap-statistics filter-list list-number begin](#) regular-expression  
[Section 14-76. show ip bgp flap-statistics filter-list list-number exclude](#)  
regular-expression

[Section 14-77. show ip bgp flap-statistics filter-list list-number |include regular-expression](#)  
[Section 14-78. show ip bgp flap-statistics quote-regexp quoted-line](#)  
[Section 14-79. show ip bgp flap-statistics quote-regexp quoted-line |begin regular-expression](#)  
[Section 14-80. show ip bgp flap-statistics quote-regexp quoted-line |exclude regular-expression](#)  
[Section 14-81. show ip bgp flap-statistics quote-regexp quoted-line |include regular-expression](#)  
[Section 14-82. show ip bgp flap-statistics regexp regular-expression](#)  
[Section 14-83. show ip bgp flap-statistics |begin regular-expression](#)  
[Section 14-84. show ip bgp flap-statistics |exclude regular-expression](#)  
[Section 14-85. show ip bgp flap-statistics |include regular-expression](#)  
[Section 14-86. show ip bgp inconsistent-as](#)  
[Section 14-87. show ip bgp inconsistent-as |begin line](#)  
[Section 14-88. show ip bgp inconsistent-as |exclude line](#)  
[Section 14-89. show ip bgp inconsistent-as |include line](#)  
[Section 14-90. show ip bgp neighbors](#)  
[Section 14-91. show ip bgp neighbors |begin line](#)  
[Section 14-92. show ip bgp neighbors |exclude line](#)  
[Section 14-93. show ip bgp neighbors |include line](#)  
[Section 14-94. show ip bgp neighbors ip-address advertised-routes](#)  
[Section 14-95. show ip bgp neighbors ip-address dampened-routes](#)  
[Section 14-96. show ip bgp neighbors ip-address flap-statistics](#)  
[Section 14-97. show ip bgp neighbors ip-address paths](#)  
[Section 14-98. show ip bgp neighbors ip-address paths line](#)  
[Section 14-99. show ip bgp neighbors ip-address received-routes](#)  
[Section 14-100. show ip bgp neighbors ip-address routes](#)  
[Section 14-101. show ip bgp paths](#)  
[Section 14-102. show ip bgp paths line](#)  
[Section 14-103. show ip bgp paths line](#)  
[Section 14-104. show ip bgp paths line](#)  
[Section 14-105. show ip bgp paths |begin line](#)  
[Section 14-106. show ip bgp paths |exclude line](#)  
[Section 14-107. show ip bgp paths |include line](#)  
[Section 14-108. show ip bgp peer-group](#)  
[Section 14-109. show ip bgp peer-group peer-group-name](#)  
[Section 14-110. show ip bgp peer-group peer-group-name summary](#)  
[Section 14-111. show ip bgp summary](#)  
[Section 14-112. show ip bgp summary |begin line](#)  
[Section 14-113. show ip bgp summary |exclude line](#)  
[Section 14-114. show ip bgp summary |include line](#)

## [Chapter 15. BGP clear Commands](#)

[Section 15-1. clear ip bgp \\*](#)  
[Section 15-2. clear ip bgp \\* soft](#)  
[Section 15-3. clear ip bgp \\* soft in](#)

[Section 15-4. clear ip bgp \\* soft out](#)  
[Section 15-5. clear ip bgp \\* soft in out](#)  
[Section 15-6. clear ip bgp AS-number](#)  
[Section 15-7. clear ip bgp AS-number soft](#)  
[Section 15-8. clear ip bgp AS-number soft in](#)  
[Section 15-9. clear ip bgp AS-number soft out](#)  
[Section 15-10. clear ip bgp AS-number soft in out](#)  
[Section 15-11. clear ip bgp neighbor-ip-address](#)  
[Section 15-12. clear ip bgp neighbor-ip-address soft](#)  
[Section 15-13. clear ip bgp neighbor-ip-address soft in](#)  
[Section 15-14. clear ip bgp neighbor-ip-address soft out](#)  
[Section 15-15. clear ip bgp neighbor-ip-address soft in out](#)  
[Section 15-16. clear ip bgp peer-group peer-group-name](#)  
[Section 15-17. clear ip bgp peer-group peer-group-name soft](#)  
[Section 15-18. clear ip bgp peer-group peer-group-name soft in](#)  
[Section 15-19. clear ip bgp peer-group peer-group-name soft out](#)  
[Section 15-20. clear ip bgp peer-group peer-group-name soft in out](#)  
[Section 15-21. clear ip bgp dampening](#)  
[Section 15-22. clear ip bgp dampening prefix mask](#)  
[Section 15-23. clear ip bgp flap-statistics](#)  
[Section 15-24. clear ip bgp flap-statistics prefix mask](#)  
[Section 15-25. clear ip bgp flap-statistics filter-list list-number](#)  
[Section 15-26. clear ip bgp flap-statistics regexp regular-expression](#)  
[Section 15-27. clear ip bgp neighbor-ip-address flap-statistics](#)

## [Chapter 16. BGP debug Commands](#)

[Section 16-1. debug ip bgp](#)  
[Section 16-2. debug ip bgp neighbor-ip-address updates](#)  
[Section 16-3. debug ip bgp neighbor-ip-address updates access-list-number](#)  
[Section 16-4. debug ip bgp dampening](#)  
[Section 16-5. debug ip bgp dampening access-list-number](#)  
[Section 16-6. debug ip bgp events](#)  
[Section 16-7. debug ip bgp keepalives](#)  
[Section 16-8. debug ip bgp in neighbor-ip-address updates](#)  
[Section 16-9. debug ip bgp in neighbor-ip-address updates access-list-number](#)  
[Section 16-10. debug ip bgp out neighbor-ip-address updates](#)  
[Section 16-11. debug ip bgp out neighbor-ip-address updates access-list-number](#)  
[Section 16-12. debug ip bgp updates](#)  
[Section 16-13. debug ip bgp updates in](#)  
[Section 16-14. debug ip bgp updates out](#)  
[Section 16-15. debug ip bgp updates access-list-number](#)  
[Section 16-16. debug ip bgp updates access-list-number in](#)  
[Section 16-17. debug ip bgp updates access-list-number out](#)

## [appendix A. RFC 1771: Border Gateway Protocol 4](#)

[BGP Attributes](#)

[BGP Path Selection](#)



[Forming a BGP Connection](#)

[Appendix B. Regular Expressions](#)

[appendix C. Route Map Logic](#)

[Form 1: permit/permit](#)

[Form 2: permit/deny](#)

[Form 3: deny/permit](#)

[Form 4: deny/deny](#)

[Index](#)

◀ PREVIOUS

NEXT ▶



# Copyright

Copyright © 2001 Cisco Press

Cisco Press logo is a trademark of Cisco Systems, Inc.

Published by:

Cisco Press

201 West 103rd Street

Indianapolis, IN 46290 USA

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

Library of Congress Cataloging-in-Publication Number: 2001086613

## Warning and Disclaimer

This book is designed to provide information about Cisco IOS Software commands for Border Gateway Protocol Version 4 (BGP-4). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The author, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

## Trademark Acknowledgments



## About the Author

William R. Parkhurst, Ph.D., CCIE #2969, is the manager of the CCIE Development group at Cisco Systems. The CCIE Development group is responsible for all new CCIE written qualification and laboratory exams. Prior to joining the CCIE team, Bill was a Consulting Systems Engineer supporting the Sprint Operation. Bill first became associated with Cisco Systems while he was a Professor of Electrical and Computer Engineering at Wichita State University. In conjunction with Cisco Systems, WSU established the first CCIE Preparation Laboratory.

## About the Technical Reviewers

Bill Wagner works as a Cisco Certified Systems Instructor for Mentor Technologies. He has 22 years of computer programming and data communication experience. He has worked for corporations and companies such as Independent Computer Consultants, Numerax, McGraw-Hill, and Standard and Poors. His teaching experience started with the Chubb Institute, Protocol Interface, Inc., and Geotrain. Currently he teaches at Mentor Technologies.

Steve Wisniewski is a Systems Engineer for Fujitsu Network Communications. Steve has authored a book titled Network Administration from Prentice Hall and has edited several other Cisco Press books. Steve resides with his wife Ellen in East Brunswick, New Jersey.

# Acknowledgments

Writing a book is never an easy endeavor. Without the help and guidance of John Kane, Editor-In-Chief, and Christopher Cleveland, Development Editor, of Cisco Press the task of writing this book would have been less enjoyable. John and Chris may have cracked the whip occasionally but it was always done with diplomacy and humor. I want to especially acknowledge my wife Debbie for her constant encouragement and for the wonderful job she did in proofreading the manuscript. The number of required corrections was minimal thanks to her efforts and attention to detail. Debbie made me look good in the eyes of my editor and for that I am thankful.





# Introduction

I have been involved with the world of networking from many directions. My experiences in education, network consulting, service provider support, and certification have shown me that there is a common thread that frustrates people in all of these arenas. That common thread is documentation. There are many factors that cause documentation to be frustrating but the most common are amount, clarity, and completeness. The amount of documentation available, especially in regards to BGP, can be overwhelming. For a person who is beginning to learn BGP, the question is "where do I begin"? There are very good books, RFCs, white papers, and command references available, but it is difficult to know where to start. The clarity of documentation depends on your personal situation. For a seasoned BGP designer the documentation may be clear and concise. To an individual preparing for a professional certification such as the CCIE the same documentation may be confusing. Even if the documentation is clear it is sometimes not complete. You may understand the words but be confused on the application. The purpose of this book is to provide a BGP handbook that is clear, concise, and complete. This book is not meant to be read from cover to cover. The way you use this book will depend on your objectives. If you are preparing for the CCIE written and lab exams then this book can be used as a laboratory guide to learn the purpose and proper use of every BGP command. If you are a network designer then this book can be used as a ready reference for any BGP command. In order to satisfy these varying audiences the structure of this book is reasonably simple. Each BGP command is illustrated using the following structure:

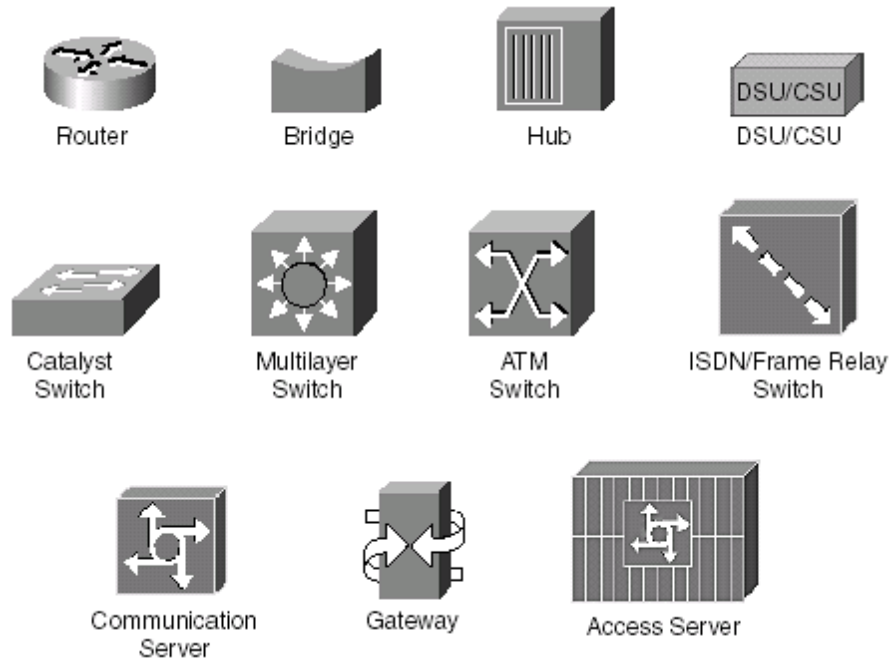
- 
- Listing of the command structure and syntax
- 
- 
- Syntax description for the command with an explanation of all command parameters
- 
- 
- The purpose of the command and the situation where the command is used
- 
- 
- The first release of the IOS in which the command appeared
- 
- 
- One or more configuration examples to demonstrate the proper use of the command
- 
- 
- Procedures and examples to verify that the command is working properly
- 
- 
- How to troubleshoot the command when things are not working as intended.
-



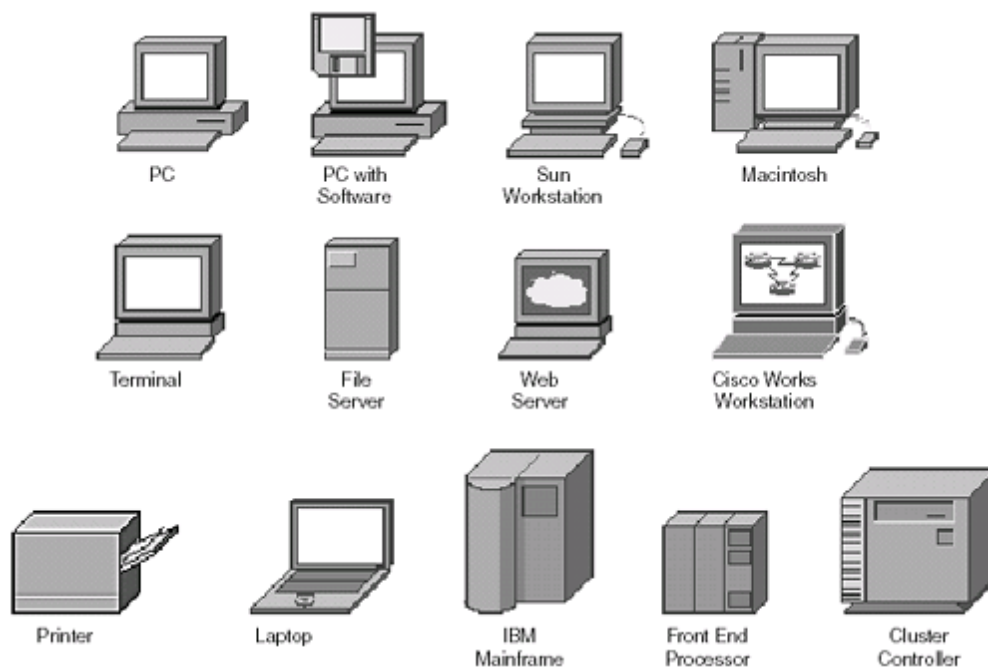


# Icons Used in This Book

Throughout the book, you will see the following icons used for networking devices:



Throughout the book, you will see the following icons used for peripherals and other devices.



Throughout the book, you will see the following icons used for networks and network connections.



## Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- 
- Vertical bars (|) separate alternative, mutually exclusive elements.
- 
- 
- Square brackets [ ] indicate optional elements.
- 
- 
- Braces { } indicate a required choice.
- 
- 
- Braces within brackets [{ }] indicate a required choice within an optional element.
- 
- 
- Boldface indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a show command).
- 
- 
- Italics indicate arguments for which you supply actual values.
-

# Chapter 1. Route Aggregation

- [Section 1-1. aggregate-address](#) address mask
- 
- [Section 1-2. aggregate-address](#) address mask [as-set](#)
- 
- [Section 1-3. aggregate-address](#) address mask [as-set advertise-map](#) route-map-name
- 
- [Section 1-4. aggregate-address](#) address mask [attribute-map](#) route-map-name
- 
- [Section 1-5. aggregate-address](#) address mask [route-map](#) route-map-name
- 
- [Section 1-6. aggregate-address](#) address mask [summary-only](#)
- 
- [Section 1-7. aggregate-address](#) address mask [suppress-map](#) route-map-name
-





# 1-1 aggregate-address address mask

Syntax Description:

- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 

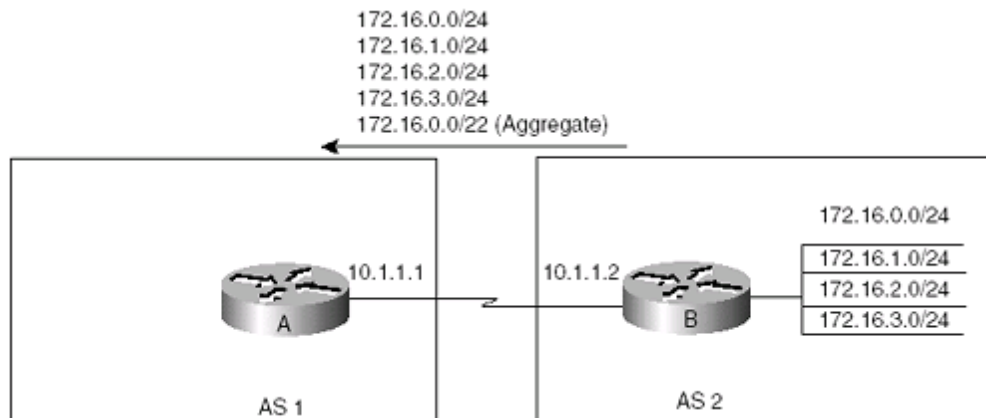
Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. This form of the aggregate-address command advertises the aggregate and all the more-specific routes that are part of the aggregate.

Cisco IOS Software Release: 10.0

## Configuration Example 1: Aggregating Local Routes

For this example, we will aggregate locally sourced routes. In [Figure 1-1](#), Router B aggregates the four networks 172.16.0.x through 172.16.3.x.

Figure 1-1. Aggregating Locally Sourced Routes



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```





## 1-2 aggregate-address address mask as-set

Syntax Description:

- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 

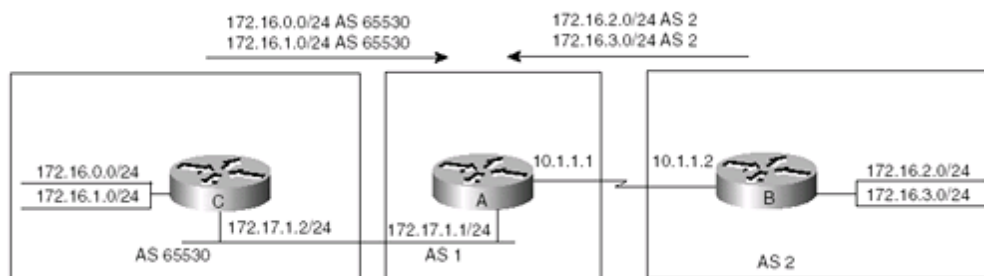
Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. Without the as-set option, the AS path information for the specific routes forming the aggregate is lost. This form of the aggregate-address command advertises the aggregate while retaining the AS path information for the more-specific routes.

Cisco IOS Software Release: 10.0

### Configuration Example: Forming an Aggregate Consisting of Prefixes from Different Autonomous Systems

In [Figure 1-4](#), Router A is learning about networks 172.16.0.0/24 and 172.16.1.0/24 from AS 65530 and networks 172.16.2.0/24 and 172.16.3.0/24 from AS 2.

**Figure 1-4. Aggregate Consists of Routes Originated in Multiple Autonomous Systems**



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```

```
neighbor 172.17.1.2 remote-as 65530
```





# 1-3 aggregate-address address mask as-set advertise-map route-map-name

Syntax Description:

- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 
- 
- route-map-name— Route map used to determine the prefixes used to form the aggregate.
- 

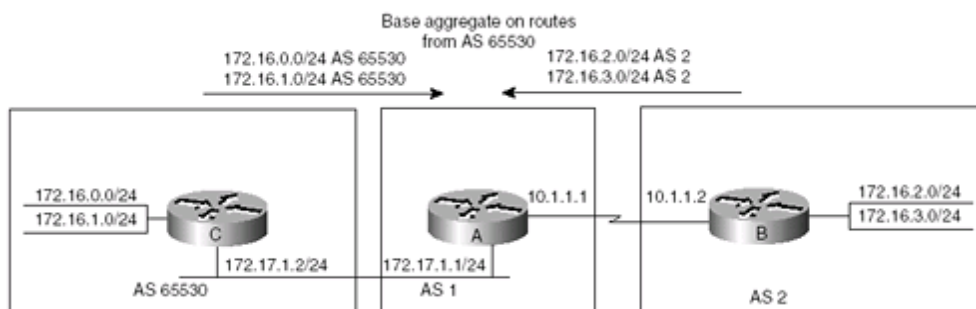
Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. Without the as-set option, the AS path information for the specific routes forming the aggregate is lost. This form of the aggregate-address command advertises the aggregate while retaining the AS path information for the more-specific routes. An advertise-map can be used to determine which AS path information is retained in the aggregate.

Cisco IOS Software Release: 10.0

## Configuration Example: Forming an Aggregate Based on a Subset of Prefixes from Different Autonomous Systems

In [Figure 1-5](#), Router A is learning about networks 172.16.0.0/24 and 172.16.1.0/24 from AS 65530 and networks 172.16.2.0/24 and 172.16.3.0/24 from AS 2. We will use the advertise-map option to base the aggregate on routes received from AS 65530.

Figure 1-5. Aggregate Based on Routes from a Specific Autonomous System







**1-4 aggregate-address** address mask **attribute-map** route-map-name



## 1-5 aggregate-address address mask route-map route-map-name

Syntax Description:

- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 
- 
- route-map-name— Route map used to modify the aggregate's attributes.
- 

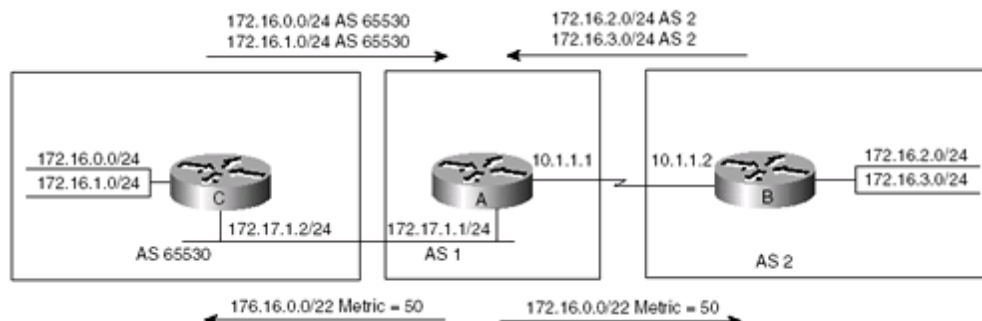
Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. This form of the aggregate-address command can be used to modify the aggregate's BGP attributes. The form of the command using the keyword route-map is equivalent to using the keyword attribute-map.

Cisco IOS Software Release: 10.0

### Configuration Example: Modifying the Aggregate's Attributes

[Figure 1-6](#) shows Router A learning two routes from Router B and two routes from Router C. Router B aggregates these four routes and modifies the aggregate's metric.

**Figure 1-6. An Attribute Map Is Used to Modify the Attributes of an Aggregate Route**



Router A

router bgp 1





## 1-6 aggregate-address address mask summary-only

Syntax Description:

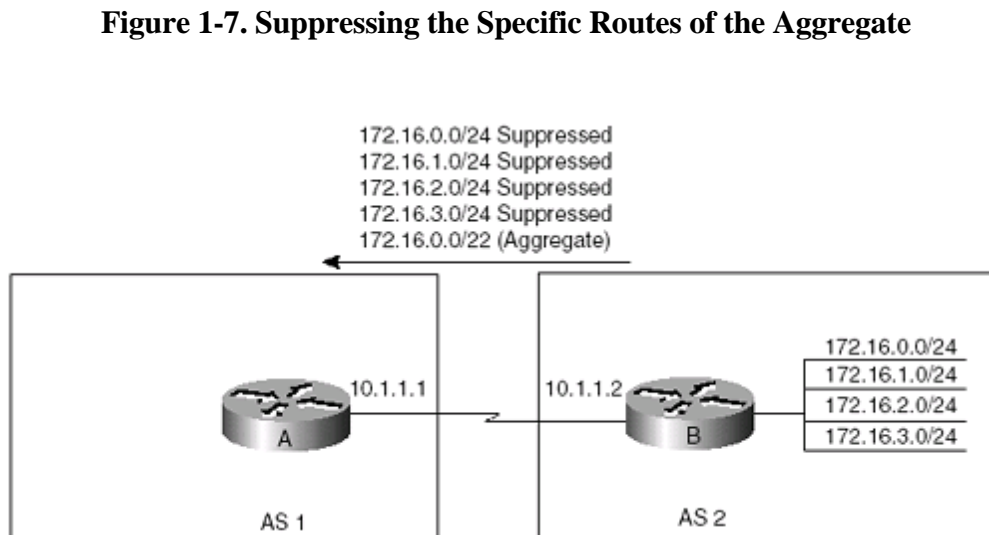
- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 

Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. This form of the aggregate-address command advertises the aggregate while suppressing all the more-specific routes.

Cisco IOS Software Release: 10.0

### Configuration Example: Advertise an Aggregate While Suppressing the More-Specific Routes

In [Figure 1-7](#), Router B is generating an aggregate for 172.16.0.0/22 while suppressing the specific routes that comprise the aggregate.



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```







## 1-7 aggregate-address address mask suppress-map route-map-name

Syntax Description:

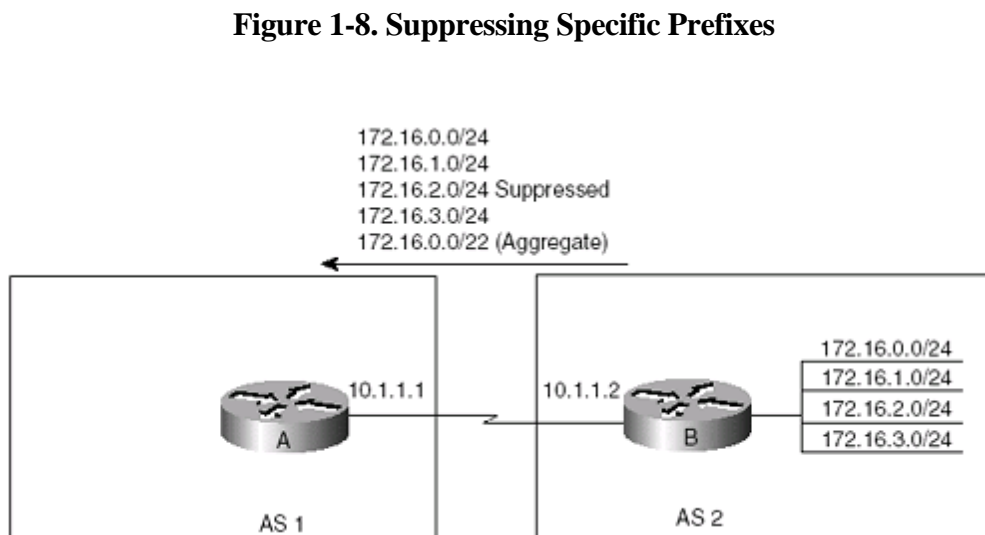
- 
- address— Aggregate IP address.
- 
- 
- mask— Aggregate mask.
- 
- 
- route-map-name— Name of the route map that is used to determine which specific prefixes will be suppressed.
- 

Purpose: To create an aggregate entry in the BGP table. An aggregate is created only if a more-specific route of the aggregate exists in the BGP table. This form of the aggregate-address command advertises the aggregate while suppressing the more-specific routes indicated by the route map.

Cisco IOS Software Release: 10.0

### Configuration Example: Suppressing a Subset of the More-Specific Routes Used to Form the Aggregate

In [Figure 1-8](#), Router B is generating an aggregate for 172.16.0.0/22 while suppressing one of the more-specific prefixes.





## Chapter 2. Auto-Summary

- [Section 2-1. auto-summary](#)
-



## 2-1 auto-summary

### Syntax Description:

This command has no arguments. auto-summary is enabled by default.

**Purpose:** When auto-summary is enabled, routes injected into BGP via redistribution are summarized on a classful boundary. A 32-bit IP address consists of a network address and a host address. The subnet mask determines the number of bits used for the network address and the number of bits used for the host address. The IP address classes have a natural or standard subnet mask, as shown in [Table 2-1](#).

Table 2-1. IP Address Classes

Class	Address Range	Natural Mask
A	1.0.0.0 to 126.0.0.0	255.0.0.0 or /8
B	128.1.0.0 to 191.254.0.0	255.255.0.0 or /16
C	192.0.1.0 to 223.255.254.0	255.255.255.0 or /24
Reserved addresses include 128.0.0.0, 191.255.0.0, 192.0.0.0, and 223.255.255.0		

When using the standard subnet mask, Class A addresses have one octet for the network, Class B addresses have two octets for the network, and Class C addresses have three octets for the network. As an example, consider the Class B address 156.26.32.1 with a 24-bit subnet mask. The 24-bit subnet mask selects three octets, 156.26.32, for the network. The last octet is the host address. If the network 156.26.32.1/24 is learned via an IGP and is then redistributed into BGP, the network is automatically summarized to the natural mask for a Class B network. The network that BGP advertises is 156.26.0.0/16. BGP is advertising that it can reach the entire Class B address space from 156.26.0.0 to 156.26.255.255. If the only network that can be reached via the BGP router is 156.26.32.0/24, BGP is advertising 254 networks that cannot be reached via this router.

auto-summary does not apply to routes injected into BGP via the network command or through IBGP or EBGP.

Cisco IOS Software Release: 10.0

### Configuration Example: Automatic Route Summarization

This example demonstrates the effect of automatic route summarization. In [Figure 2-1](#), static routes, connected routes, and routes learned via OSPF are being redistributed into BGP as shown in the following configurations. The redistributed routes are summarized to a classful boundary by BGP because auto-summary is enabled by default.





# Chapter 3. BGP-Specific Commands

- [Section 3-1. bgp always-compare-med](#)  
•
- [Section 3-2. bgp bestpath as-path ignore](#)  
•
- [Section 3-3. bgp bestpath med confed](#)  
•
- [Section 3-4. bgp bestpath med missing-as-worst](#)  
•
- [Section 3-5. bgp client-to-client reflection](#)  
•
- [Section 3-6. bgp cluster-id 32-bit\\_id](#)  
•
- [Section 3-7. bgp confederation identifier AS-number](#)  
•
- [Section 3-8. bgp confederation peers 1\\_or\\_more\\_AS-numbers](#)  
•
- [Section 3-9. bgp dampening](#)  
•
- [Section 3-10. bgp dampening half-life](#)  
•
- [Section 3-11. bgp dampening half-life reuse suppress max-suppress-time](#)  
•
- [Section 3-12. bgp dampening route-map route-map-name](#)  
•
- [Section 3-13. bgp default local-preference local-preference](#)  
•
- [Section 3-14. bgp deterministic-med](#)  
•
- [Section 3-15. bgp fast-external-fallover](#)  
•
- [Section 3-16. bgp log-neighbor-changes](#)







## 3-1 bgp always-compare-med

Syntax Description:

This command has no arguments.

Purpose: If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes in the following order:

1.
  1. Ignore a route if the next hop is not known.
  - 1.
  - 2.
2. Ignore IBGP routes that are not synchronized.
  - 2.
  - 3.
3. Prefer the route with the largest weight.
  - 3.
  - 4.
4. Prefer the route with the largest local preference.
  - 4.
  - 5.
5. Prefer the route that was locally originated.
  - 5.
  - 6.
6. Prefer the route with the shortest AS path.
  - 6.
6. If you're using `bgp bestpath as-path ignore`, skip this step. When you use the `as-set` option for aggregated routes, `as-set` counts as 1 regardless of the number of AS entries in the set. Confederation sub-AS numbers are not used to determine the AS-path length.
  - 6.
  - 7.
7. Prefer the route with the lowest origin (IGP < EGP < Incomplete).
  - 7.
  - 8.
8. Prefer the route with the lowest MED.
  - 8.





## 3-2 bgp bestpath as-path ignore

Syntax Description:

This command has no arguments.

Purpose: If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes in the following order:

1.
  1. Ignore a route if the next hop is not known.
  - 1.
  - 2.
2. Ignore IBGP routes that are not synchronized.
  - 2.
  - 3.
3. Prefer the route with the largest weight.
  - 3.
  - 4.
4. Prefer the route with the largest local preference.
  - 4.
  - 5.
5. Prefer the route that was locally originated.
  - 5.
  - 6.
6. Prefer the route with the shortest AS path.
  - 6.
  6. If you're using bgp bestpath as-path ignore, skip this step. When you use the as-set option for aggregated routes, as\_set counts as 1 regardless of the number of AS entries in the set. Confederation sub-AS numbers are not used to determine the AS-path length.
  - 6.
  - 7.
7. Prefer the route with the lowest origin (IGP < EGP < Incomplete).
  - 7.
  - 8.
8. Prefer the route with the lowest MED. This comparison is only between routes advertised by the same external AS.
  - 8.







## 3-3 bgp bestpath med confed

Syntax Description:

This command has no arguments.

Purpose: If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes in the following order:

1.
  1. Ignore a route if the next hop is not known.
  - 1.
  - 2.
2. Ignore IBGP routes that are not synchronized.
  - 2.
  - 3.
3. Prefer the route with the largest weight.
  - 3.
  - 4.
4. Prefer the route with the largest local preference.
  - 4.
  - 5.
5. Prefer the route that was locally originated.
  - 5.
  - 6.
6. Prefer the route with the shortest AS path.
  - 6.
  6. If you're using `bgp bestpath as-path ignore`, skip this step. When you use the `as-set` option for aggregated routes, `as_set` counts as 1 regardless of the number of AS entries in the set. Confederation sub-AS numbers are not used to determine the AS-path length.
  - 6.
  - 7.
7. Prefer the route with the lowest origin (IGP < EGP < Incomplete).
  - 7.
  - 8.
8. Prefer the route with the lowest MED. This comparison is only between routes advertised by the same external AS.
  - 8.





## 3-4 bgp bestpath med missing-as-worst

Syntax Description:

This command has no arguments.

Purpose: If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes in the following order:

- 1.
1. Ignore a route if the next hop is not known.
  - 1.
  - 2.
2. Ignore IBGP routes that are not synchronized.
  - 2.
  - 3.
3. Prefer the route with the largest weight.
  - 3.
  - 4.
4. Prefer the route with the largest local preference.
  - 4.
  - 5.
5. Prefer the route that was locally originated.
  - 5.
  - 6.
6. Prefer the route with the shortest AS path.
  - 6.
6. If using bgp bestpath as-path ignore, skip this step. When using the as-set option for aggregated routes, as\_set counts as 1 regardless of the number of AS entries in the set. Confederation sub-AS numbers are not used to determine the AS-path length.
  - 6.
  - 7.
7. Prefer the route with the lowest origin (IGP < EGP < Incomplete).
  - 7.
  - 8.
8. Prefer the route with the lowest MED. This comparison is only between routes advertised by the same external AS.
  - 8.





## 3-5 bgp client-to-client reflection

Syntax Description:

This command has no arguments.

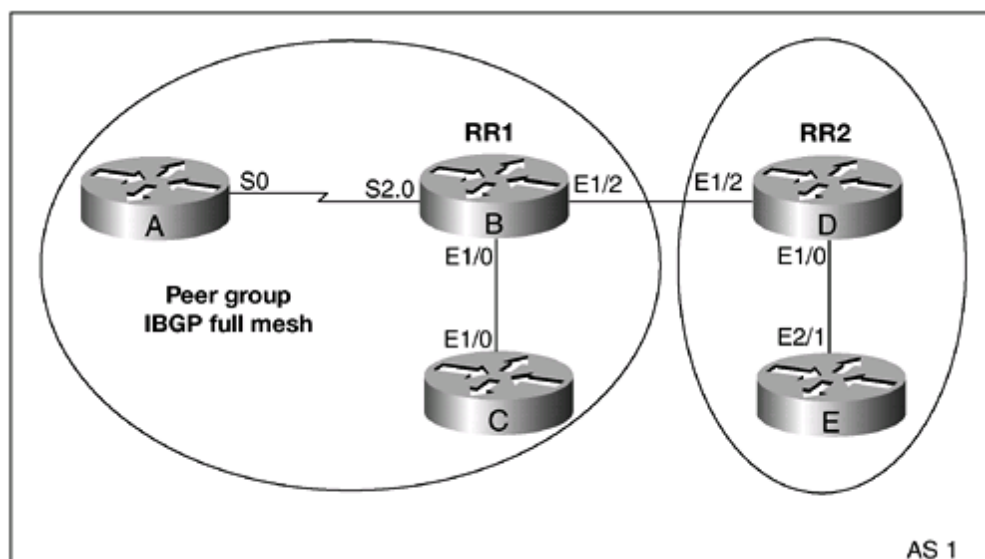
Purpose: When a router is configured as a route reflector, client-to-client reflection is enabled by default. If the route reflector's clients are fully meshed, client-to-client reflection can be disabled on the route reflector using the no form of the command.

Cisco IOS Software Release: 11.1

### Configuration Example: Route Reflectors and Peer Groups

An obvious question is why use a route reflector if the clients have a full IBGP mesh? If client-to-client reflection is enabled, the clients cannot be part of a peer group. [Figure 3-5](#) demonstrates a situation in which we want to scale the IBGP connections using route reflectors, but we also want Routers A, B, and C to be part of the same peer group. Therefore, client-to-client reflection needs to be disabled on Router B. The total number of IBGP connections is still reduced, even though the clients are fully meshed. Routers B and D need to be configured as route reflectors. If B and D were not route reflectors, a full mesh would be required between all five routers. A route reflector is set up by configuring Routers A and C as route reflector clients on Router B. But because the clients are fully meshed, route reflection can be disabled on Router B.

**Figure 3-5. Route Reflectors and Peer Groups**









### 3-6 bgp cluster-id 32-bit\_id

Syntax Description:

32-bit\_id— Route reflector cluster ID entered as either a 32-bit number or an IP address.

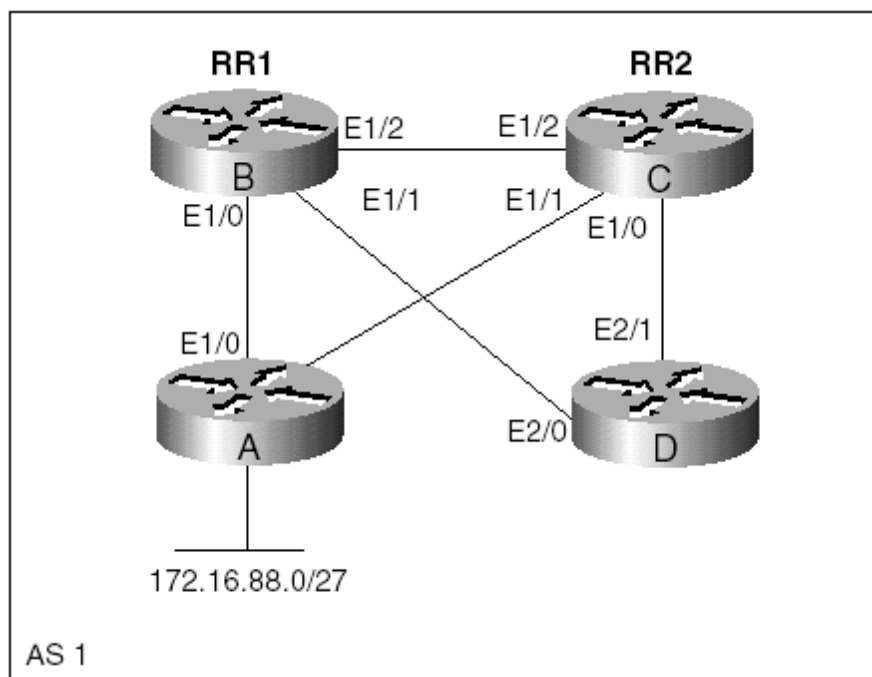
Purpose: EBGp loop detection is based on the AS-path attribute. If a BGP router receives an update from an EBGp peer containing the local AS number, the update is ignored. The AS-path method of loop detection does not work when you're using route reflectors, because the reflector and the clients are in the same AS. A route reflector generates an originator ID assigned from the router ID. When you're using multiple route reflectors, the route reflectors in the same cluster must be configured with the same ID, called the cluster ID. A cluster consists of the router reflectors and their clients. The cluster IDs contained in the cluster list are used for loop detection within the local AS when route reflectors are being used.

Cisco IOS Software Release: 11.0

#### Configuration Example: Redundant Route Reflectors

[Figure 3-6](#) shows a cluster consisting of two clients and two route reflectors. Two route reflectors are configured for redundancy and therefore must have the same cluster ID.

**Figure 3-6. Redundant Route Reflectors**





## 3-7 bgp confederation identifier AS-number



### 3-8 bgp confederation peers 1\_or\_more\_AS-numbers

Syntax Description:

- 
- AS-number— AS number used with EBGP neighbors.
- 
- 
- 1\_or\_more\_AS-numbers— AS number(s) of directly connected peers that are in a different sub-AS.
- 

Purpose: IBGP neighbors do not propagate routing information learned from one IBGP neighbor to another IBGP neighbor. If you are running IBGP, every IBGP speaker must have a connection to every other IBGP speaker in the AS. This becomes a scaling problem as the number of IBGP speakers increases. The number of IBGP connections for n speakers is  $[n(n-1)]/2$ . [Table 3-1](#) lists the number of connections needed for two to ten IBGP speakers.

Table 3-1. IBGP Connections Needed for a Full Mesh

Number of IBGP Speakers	Number of Connections
2	1
3	3
4	6
5	10
6	15
7	21
8	28
9	36
10	45

A confederation is one technique used to overcome the scaling issue with IBGP. The AS is divided into multiple subautonomous systems. Within a confederation sub-AS, a full IBGP mesh is required. BGP connections between confederations behave like EBGP peers, but they exchange routing information as if they were using IBGP. This means that the BGP attributes next hop, metric, and local preference are preserved. To an EBGP neighbor, the confederation appears as a single AS.



## 3-9 bgp dampening



## 3-10 bgp dampening half-life

## 3-11 bgp dampening half-life reuse suppress max-suppress-time



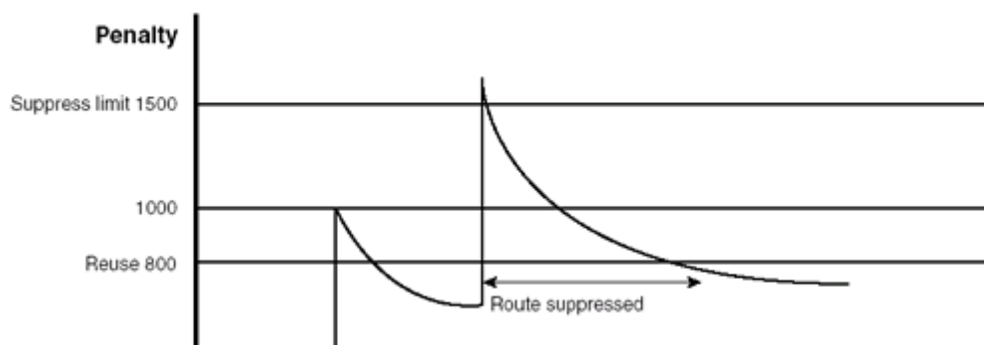
### 3-12 bgp dampening route-map route-map-name

#### Syntax Description:

- 
- half-life— Time for the penalty to decrease to one-half of its current value. The range of values is 1 to 45 minutes. The default is 15 minutes.
- 
- 
- reuse— When the penalty for a suppressed route decays below the reuse value, the route becomes unsuppressed. The range of values is 1 to 20,000. The default value is 750.
- 
- 
- suppress— When the penalty for a route exceeds the suppress value, the route is suppressed. The range of values is 1 to 20,000. The default value is 2000.
- 
- 
- max-suppress-time— Maximum time that a dampened route is suppressed. The range of values is 1 to 255 minutes. The default value is four times the half-life value.
- 
- 
- route-map-name— Name of the route map used to select prefixes and dampening parameters.
- 

Purpose: A route flap occurs when a prefix transitions from the up to the down state. When the prefix goes from up to down, BGP must send a WITHDRAWN message. When the prefix goes from down to up, BGP sends an UPDATE message. If the prefix is constantly flapping, this can cause high CPU utilization while the BGP routes are converging. Additionally, if you are redistributing BGP into your IGP, the flapping route can cause instability in the IGP. Dampening is a method to control the effect of a flapping route. The mechanics of route dampening are illustrated in [Figure 3-8](#).

**Figure 3-8. BGP Dampening Parameters and Operation**







### 3-13 bgp default local-preference local-preference

Syntax Description:

- 
- local-preference— Value to use as the default local preference. The range of values is 0 to 4294967295.
- 

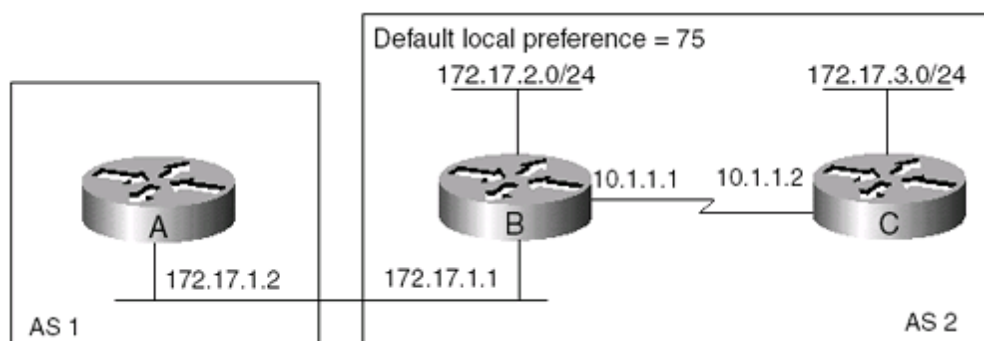
Purpose: Routes originating on a BGP router that are advertised within the local AS have a default local preference of 100. This command allows you to set the local preference of locally advertised routes to a value other than the default value of 100. The new value for the local preference is applicable only within the local autonomous system.

Cisco IOS Software Release: 10.0

#### Configuration Example: Default Local Preference

Router B in [Figure 3-10](#) is modifying the default local preference. The value of the local preference on Router B affects only routes advertised to Router C, because Routers B and C are in the same autonomous system.

**Figure 3-10. Modifying Local Preference for Locally Advertised Routes**



Router A

```
interface FastEthernet0
```

```
ip address 172.17.1.2 255.255.255.0
```

```
!
```

```
router bgp 1
```





## 3-14 bgp deterministic-med

Syntax Description:

This command has no arguments.

**Purpose:** If `bgp always-compare-med` is enabled, the metric or MED is compared to every path regardless of the neighbor AS. Without `bgp deterministic-med` or `bgp always-compare-med`, the order of the paths can make a difference when selecting the best path. When you're using `bgp deterministic-med`, the router sorts the paths based on neighbor AS and MED so that the paths are sorted the same way every time. This produces a deterministic best-path selection.

Cisco IOS Software Release: 12.0



## 3-15 bgp fast-external-fallover

Syntax Description:

This command has no arguments.

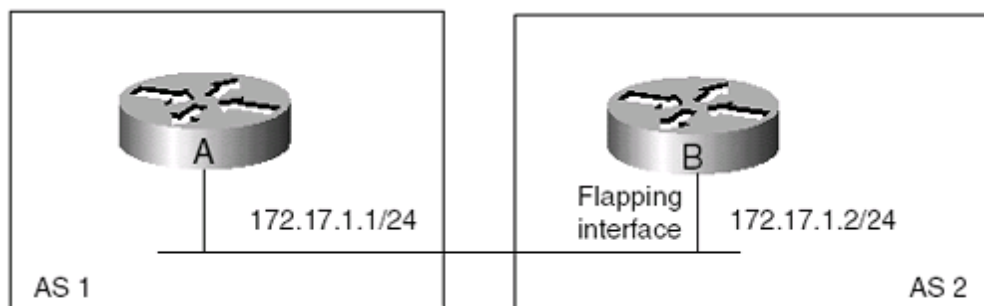
Purpose: Fast external fallover is enabled by default. When an interface that is used for a BGP connection goes down, the BGP session is immediately terminated. If the interface is flapping, instability can be caused, because the neighbors will constantly be transitioning between the idle and established states. There will also be a flood of BGP UPDATE and WITHDRAWN messages. If you have a flapping interface, use the no form of this command.

Cisco IOS Software Release: 11.0

### Configuration Example: Demonstration of Fast External Fallover

This configuration example demonstrates using the `bgp fast-external-fallover` command. In [Figure 3-11](#), the Ethernet interface on Router B is flapping. We will investigate the result of using both `fast-external-fallover` and no `fast-external-fallover`.

**Figure 3-11. A Flapping Interface Can Cause Instability**



Router A

```
router bgp 1
```

```
neighbor 172.17.1.2 remote-as 2
```

---

Router B

```
router bgp 2
```





## 3-16 bgp log-neighbor-changes

Syntax Description:

This command has no arguments.

Purpose: To enable the logging of changes in a BGP neighbor's status. If the UNIX syslog facility is enabled, messages can be sent to a UNIX host running the syslog daemon. If you are not using the UNIX syslog facility, the status change messages are stored in the router's internal buffer. Events that are logged include the following:

- 
- BGP protocol initialization
- 
- 
- No memory for path entry
- 
- 
- No memory for attribute entry
- 
- 
- No memory for prefix entry
- 
- 
- No memory for aggregate entry
- 
- 
- No memory for dampening info
- 
- 
- No memory for BGP updates
- 
- 
- BGP notification received
- 
- 
- Erroneous BGP update received
-







### 3-17 bgp router-id ip-address

Syntax Description:

- 
- ip-address— IP address to use for the BGP router ID.
- 

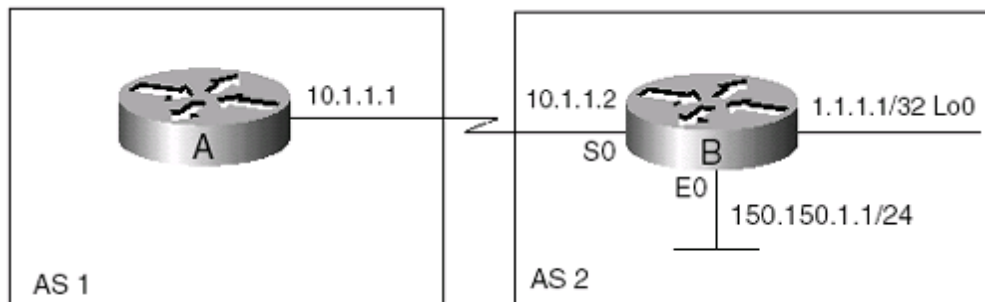
Purpose: To explicitly set the BGP router ID. BGP normally uses the highest IP address assigned to an interface as the router ID. If loopback interfaces are used, the BGP router ID is the highest address assigned to a loopback interface, regardless of the IP addresses assigned to any physical interface.

Cisco IOS Software Release: 10.0

### Configuration Example: BGP Router IDs

The scenario in [Figure 3-12](#) is used to demonstrate the three possibilities for determining a BGP router ID. Initially, configure Router B without using loopback interfaces.

**Figure 3-12. Scenario Used to Demonstrate BGP Router IDs**



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```

---

Router B

```
interface ethernet 0
```

```
ip address 150.150.1.1 255.255.255.0
```



# Chapter 4. Default Information

- [Section 4-1. default-information originate](#)
- 
- [Section 4-2. default-metric](#) metric
-



## 4-1 default-information originate

Syntax Description:

This command has no arguments.

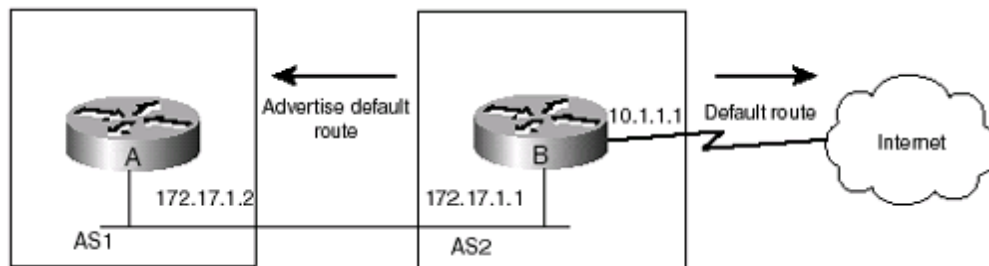
Purpose: To allow BGP to advertise the default route 0.0.0.0. A default route can also be advertised on a per-neighbor basis. See [sections 8-3](#) and [8-4](#).

Cisco IOS Software Release: 10.0

### Configuration Example: BGP Default Route Advertisement

Router B in [Figure 4-1](#) needs to be configured to advertise a default route to Router A via BGP. The configuration requires a combination of commands that we will step through to illustrate the process.

**Figure 4-1. Using BGP to Advertise a Default Route**



Router A

```
router bgp 1
```

```
neighbor 172.17.1.1 remote-as 2
```

Router B

```
router bgp 2
```

```
default-information originate
```

```
neighbor 172.17.1.2 remote-as 1
```





## 4-2 default-metric metric

Syntax Description:

- 
- metric— A metric or MED to assign to redistributed routes. The value range is 1 to 4,294,967,295.
- 

Purpose: To assign a metric or MED to routes that are redistributed into BGP. There are three methods for assigning the metric or MED for redistributed routes. The first is to not assign a metric when redistributing routes into BGP. If a metric value is not assigned, the value 0 is applied to the metric for redistributed routes. This occurs when you use the redistribute command with no metric assignment (see [section 10-1](#)). The second method is to assign a metric or MED value when redistributing a protocol into BGP (see [section 10-2](#)). The third method uses the default-metric command to assign a metric or MED to redistributed routes that have not had their metric value assigned by the redistribute command:

- 
- redistribute ospf 1— Assigns a metric of 0 to OSPF routes.
- 
- 
- redistribute ospf 1 metric 5— Assigns a metric of 5 to OSPF routes.
- 
- 
- redistribute ospf 1 metric 5
- redistribute static
- 
- default-metric 10— Assigns a metric of 5 to OSPF routes and a metric of 10 to static routes.
- 

Cisco IOS Software Release: 10.0

### Configuration Example: Assigning Metrics to Redistributed Routes

In [Figure 4-2](#), Router B is redistributing OSPF and static routes into BGP. The OSPF routes are assigned a metric of 5 using the redistribute ospf 1 metric 5 command. The static routes are assigned a metric of 0 because we are not using the default-metric command or assigning a metric with the redistribute command.

**Figure 4-2. Assigning Metrics to Redistributed Routes**







# Chapter 5. BGP Administrative Distance

- [Section 5-1. distance](#) admin-distance ip-source-address ip-address-mask
- 
- [Section 5-2. distance](#) admin-distance ip-source-address ip-address-mask ip-access-list-number
- 
- [Section 5-3. distance bgp](#) external internal local
-

## **5-1 distance** admin-distance ip-source-address ip-address-mask



## 5-2 distance admin-distance ip-source-address ip-address-mask ip-access-list-number

### Syntax Description:

- 
- admin-distance— Administrative distance assigned to learned routes. The value range is 1 to 255.
- 
- 
- ip-source-address— IP address of the source of the routing update.
- 
- 
- ip-address-maskl— Address mask for the source of the routing update.
- 
- 
- ip-access-list-number— Optional standard IP access list used to apply the administrative distance to selected routes.
- 

Purpose: To modify the administrative distance of BGP routes. When a particular route is learned via multiple routing protocols, the administrative distance is used to select the best route. The lower administrative distance is preferred. The administrative distances used for the IP routing protocols are as follows:

- 
- Connected— 0
- 
- 
- Static— 1
- 
- 
- EBGp— 20
- 
- 
- EIGRP— 90
- 
- 
- IGRP— 100
- 
-





## 5-3 distance bgp external internal local

Syntax Description:

- 
- external— Routes learned via EIGRP.
- 
- 
- internal— Routes learned via IBGP
- 
- 
- local— Routes entered into the BGP table via the aggregate-address command.
- 

Defaults: external 20, internal 200, local 200

Purpose: To modify the administrative distance of BGP routes. When a particular route is learned via multiple routing protocols, the administrative distance is used to select the best route. The lower administrative distance is preferred. The administrative distances used for the IP routing protocols are:

- 
- Connected— 0
- 
- 
- Static— 1
- 
- 
- EIGRP— 20
- 
- 
- EIGRP— 90
- 
- 
- IGRP— 100
- 
- 
- OSPF— 110
- 
-





# Chapter 6. BGP Route Filtering

- [Section 6-1. distribute-list](#)
-

## 6-1 distribute-list

Purpose: This form of the distribute-list command works only with Interior Gateway Protocols. Even though this command appears as a BGP router configuration option, do not use this command when configuring BGP. Use the neighbor {ip-address | peer-group} distribute-list {in | out} command described in [sections 8-6](#) and [8-7](#).

# Chapter 7. BGP Maximum Paths

- [Section 7-1. maximum-paths](#) number-of-paths
-



## 7-1 maximum-paths number-of-paths

Syntax Description:

- 
- number-of-paths— Number of BGP learned paths to the same destination that will be installed in the IP routing table. The value can be 1 to 6.
- 

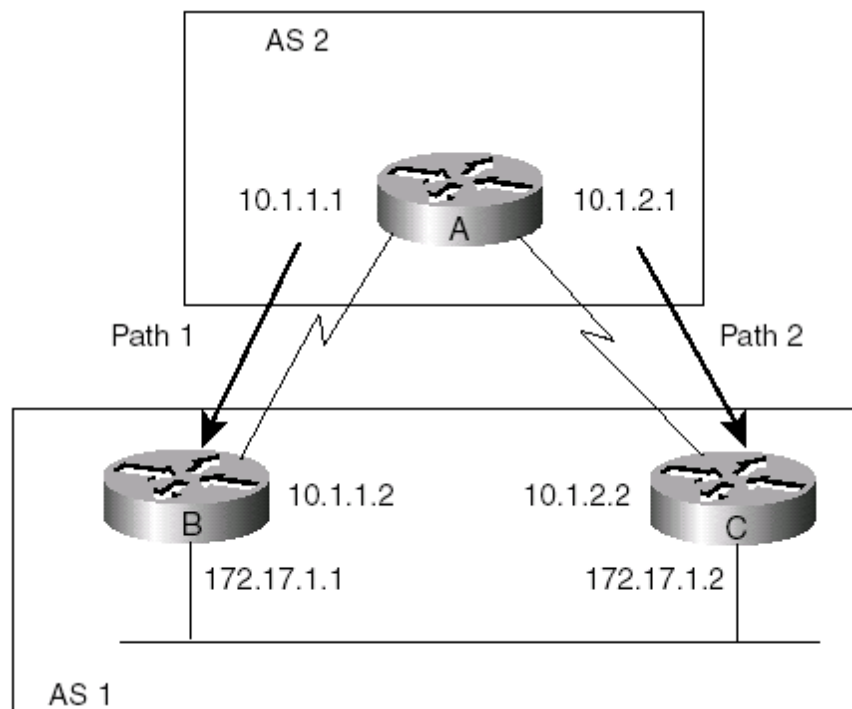
Purpose: By default, BGP installs only the best path to a destination in the IP routing table. The maximum-paths command allows up to six paths to the same destination to be installed in the IP routing table.

IOS Release: 11.2

### Configuration Example

In [Figure 7-1](#), Router A is learning two paths to network 172.17.1.x via EBGP. By default, BGP will install only one of these paths in the IP routing table. If all the attributes of the paths are equal, such as MED, Local Preference, and Weight, the route that will be installed is the one learned from the router with the lowest router ID. Initially, the routers will be configured without using the maximum-paths command, as shown in the following listing. This is done to demonstrate that only one route to 172.17.1.0 will be installed.

**Figure 7-1. Configuration Used to Demonstrate the maximum-paths Command**









# Chapter 8. Neighbor Configuration

- [Section 8-1. neighbor { ip-address | peer-group-name } advertise-map route-map-name1 non-exist-map route-map-name2](#)
- 
- [Section 8-2. neighbor { ip-address | peer-group-name } advertisement-interval seconds](#)
- 
- [Section 8-3. neighbor { ip-address | peer-group-name } default-originate](#)
- 
- [Section 8-4. neighbor { ip-address | peer-group-name } default-originate route-map route-map-name](#)
- 
- [Section 8-5. neighbor { ip-address | peer-group-name } description text](#)
- 
- [Section 8-6. neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name prefix-list-name } in](#)
- 
- [Section 8-7. neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name | prefix-list-name } out](#)
- 
- [Section 8-8. neighbor { ip-address | peer-group-name } ebgp-multihop](#)
- 
- [Section 8-9. neighbor { ip-address | peer-group-name } ebgp-multihop maximum-hop-count](#)
- 
- [Section 8-10. neighbor { ip-address | peer-group-name } filter-list as-path-list-number in](#)
- 
- [Section 8-11. neighbor { ip-address | peer-group-name } filter-list as-path-list-number out](#)
- 
- [Section 8-12. neighbor { ip-address | peer-group-name } filter-list as-path-list-number weight weight](#)
- 
- [Section 8-13. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit](#)
- 
- [Section 8-14. neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit warning-only](#)
-





## **8-1 neighbor { ip-address | peer-group-name } advertise-map route-map-name1 non-exist-map route-map-name2**

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- route-map-name1— Route map that identifies the secondary prefix to advertise only if the primary prefix referenced by route-map-name2 disappears.
- 
- 
- route-map-name2— Route map that identifies the primary prefix to advertise. If this route disappears, the secondary prefix referenced by route-map-name1 is advertised.
- 

Purpose: The primary prefix referenced by route-map-name2 is advertised to BGP peers if the prefix is in the BGP table. If the network is directly connected, the network or redistribute connected command can be used to place the primary network in the BGP table. If the primary prefix is learned via an IGP, the redistribute IGP command installs the prefix in the BGP table. The prefix may be learned from a BGP peer and is automatically placed in the BGP table. The primary prefix is advertised until it disappears from the BGP table. This can happen if the network goes down or if the advertisement for this network is no longer being received. When the route disappears, the prefix referenced by route-map-name1 is advertised. If the primary network reappears in the BGP table, it is again advertised, and the secondary prefix is suppressed.

Cisco IOS Software Release: 12.0

### **Configuration Example: Advertise the Primary Route While Suppressing the Secondary Route**

The goal of this example is to illustrate the mechanics of the neighbor advertise-map command. In [Figure 8-1](#), Router A advertises network 156.26.32.0/24 if the route is up. If network 156.26.32.0/24 goes down, network 144.223.8.0/24 is advertised.

**Figure 8-1. Illustration of the neighbor advertise-map Command**





## 8-2 neighbor { ip-address | peer-group-name } advertisement-interval seconds

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- seconds— 0 to 600.
- 

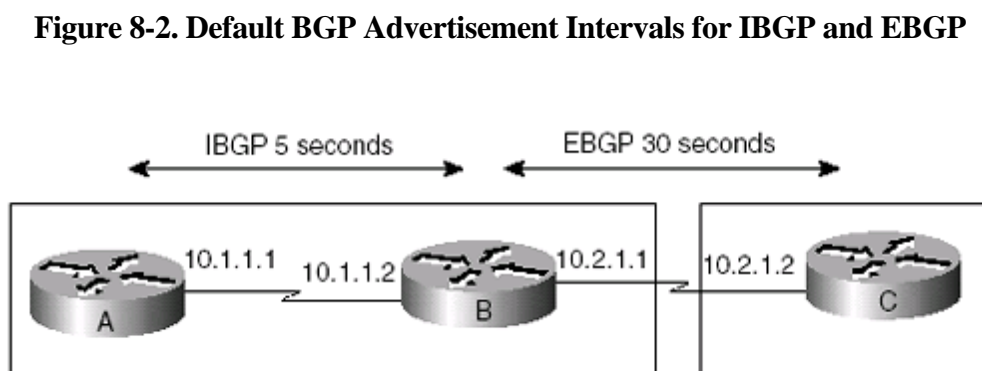
Defaults: IBGP 5 seconds. EBGP 30 seconds.

Purpose: To set the minimum interval between the sending of Border Gateway Protocol (BGP) routing updates. To restore the default setting, use the no form of this command. When a route that is being advertised by BGP changes, BGP sends either an UPDATE or WITHDRAWN message. If an advertised route is flapping, usually caused when an interface is unstable, a flood of UPDATE and WITHDRAWN messages occurs. One method to control the flooding of BGP messages is to set a minimum advertisement interval. With the default value of 30 seconds for EBGP neighbors, BGP routing updates are sent only every 30 seconds, even if a route is flapping many times during this 30-second interval. BGP dampening can also be used to control the effects of flapping routes (see [sections 3-7](#), [3-8](#), and [3-9](#)).

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Default Advertisement Interval

[Figure 8-2](#) illustrates the default advertisement interval for IBGP and EBGP connections. The default advertisement interval is automatically set when the initial neighbor relationship is established.









## 8-3 neighbor { ip-address | peer-group-name } default-originate

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

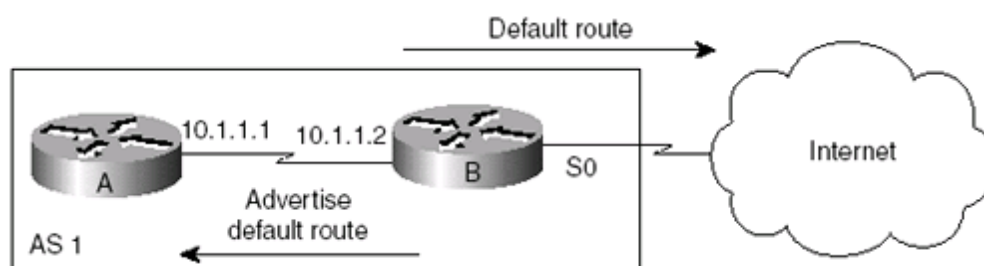
Purpose: Every router should have a default route that is used to forward packets to networks that are not in the local IP routing table. One method for ensuring that every router has a default route is to configure a static route on every router to establish the default route. Another method is to create one default route and advertise this route to the BGP neighbors. The router owning the default route can advertise it through BGP using the default-originate form of the neighbor command. Using this form is not recommended, because the router always advertises the default route, even if the router does not have a default route or if the network to the default route is down.

Cisco IOS Software Release: 11.0. Extended access lists are permitted in Release 12.0.

### Configuration Example 1: Single Default Route

[Figure 8-4](#) shows an autonomous system that has a connection to the Internet from Router B. The network directly connected to the Internet is to be used as the default route for the autonomous system. A static default route could be used on every router in the AS, but this is not the preferred method. These static routes require a high degree of maintenance. If the default route on Router B changes, every static route on every router in the AS needs to be changed. The preferred method is to dynamically propagate the default route attached to Router B throughout the AS. The following configuration contains the necessary instructions to enable Router B to propagate the default route.

**Figure 8-4. Single Default Route Advertisement**







## 8-4 neighbor { ip-address | peer-group-name } default-originate route-map route-map-name

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- route-map-name— Name of the route map.
- 

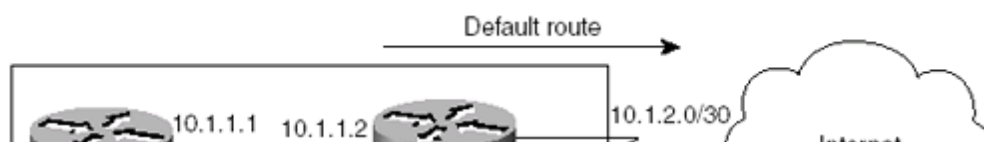
Purpose: Every router should have a default route that is used to forward packets to networks that are not in the local IP routing table. One method for ensuring that every router has a default route is to configure a static route on every router to establish the default route. Another method is to create one default route and advertise this route to the BGP neighbors. The router owning the default route can advertise it through BGP using the default-originate route-map form of the neighbor command. Using this form is recommended, because the router advertises the default route only if the condition of the specified route map is satisfied. The condition that is typically used is whether or not the default network is up.

Cisco IOS Software Release: 11.0. Extended access lists are permitted in Release 12.0.

### Configuration Example: Conditional Default Route Advertisement

[Figure 8-6](#) shows an autonomous system that has a connection to the Internet through Router B. The network directly connected to the Internet, 10.1.2.0/30, is to be used as the default route for the autonomous system. A static default route could be used on every router in the AS, but this is not the preferred method. These static routes require a high degree of maintenance. If the default route on Router B changes, every static route on every router in the AS needs to be changed. The preferred method is to dynamically propagate the default route attached to Router B throughout the AS. The following configuration contains the necessary instructions to enable Router B to propagate the default route only if the default route exists.

**Figure 8-6. Conditionally Advertising a Default Route**







## 8-5 neighbor { ip-address | peer-group-name } description text

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- text— Line describing the neighbor (1 to 80 characters).
- 

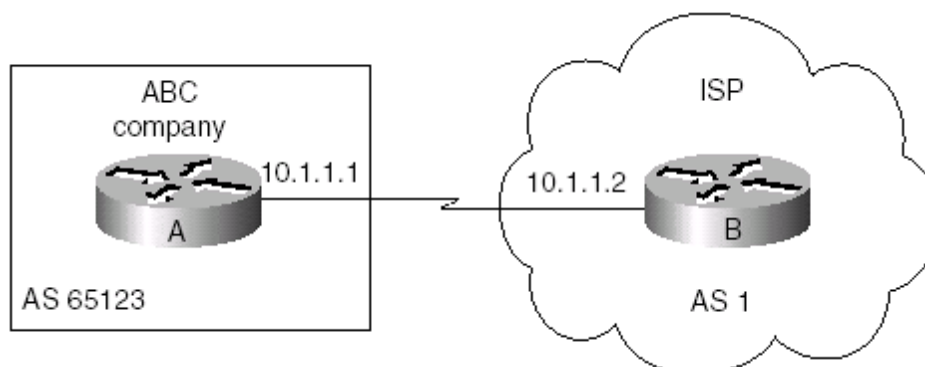
**Purpose:** The description option performs a function that is similar to a comment in a software program. This function simply helps the reader determine the purpose of the code or, in the BGP case, the identification of the neighbor. Adding a neighbor description to a BGP configuration does not affect the operation of BGP. The description should convey useful information that can be quickly used to identify neighbors. For simple scenarios that have few neighbors, the description option has limited usefulness. For an ISP that has multiple neighbor relationships, however, you can use the description command to identify a neighbor without having to memorize its IP address.

Cisco IOS Software Release: 11.3

### Configuration Example: Identifying a BGP Neighbor

[Figure 8-7](#) shows an ISP and one of its neighbor connections. The ISP uses the description option to quickly identify the customers associated with each BGP connection.

**Figure 8-7. Using the Description Option to Identify a BGP Neighbor**









## 8-6 neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name prefix-list-name } in

Syntax Description:

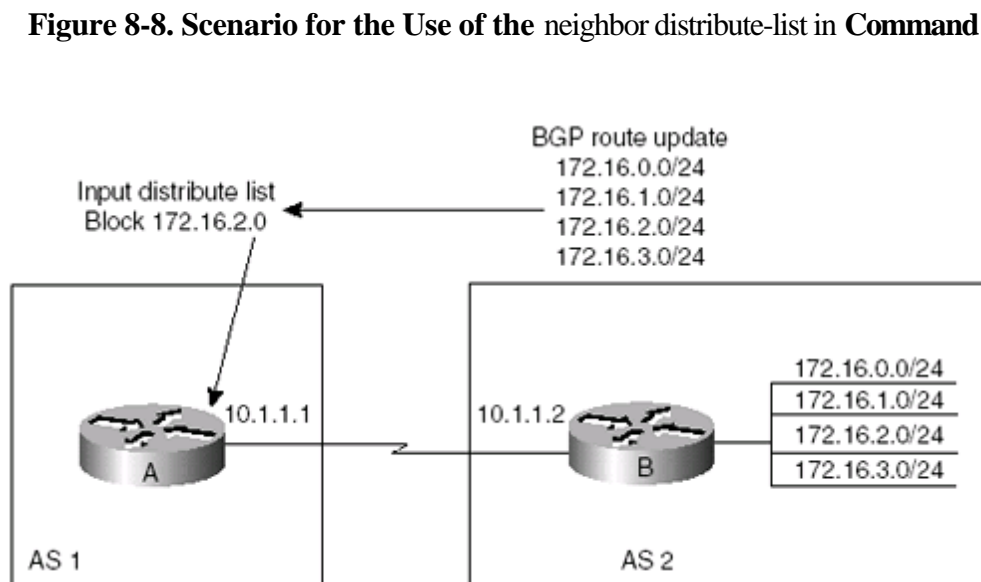
- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- ip-access-list-number-or-name— Standard, extended, or named IP access list number.
- 

Purpose: To filter incoming route updates from a particular BGP neighbor. Only one distribute list can be used per neighbor. The operation of the input distribute list is identical for both IBGP and EBGP neighbors.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0, support for named access lists was added in Release 11.2, and prefix list support was added in Release 12.0.

### Configuration Example 1: Block a Particular Route

In [Figure 8-8](#), Router B is advertising four network prefixes to Router A. Router A filters the route update from Router B in order to reject the 172.16.2.0 network. Loopbacks are used on Router B to simulate the advertised networks, as shown in the configuration.







## 8-7 neighbor { ip-address | peer-group-name } distribute-list { ip-access-list-number-or-name | prefix-list-name }out

Syntax Description:

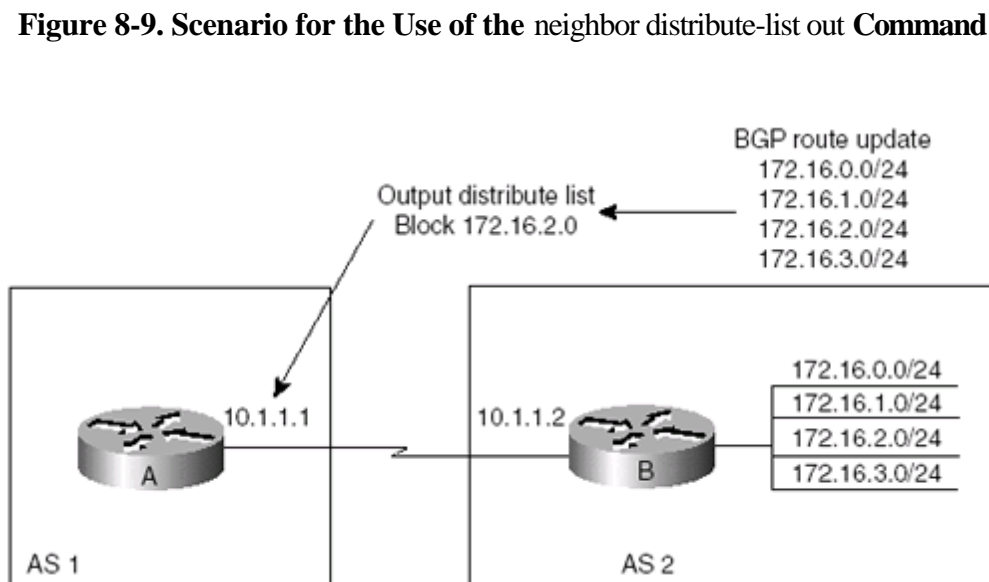
- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- ip-access-list-number-or-name— Standard, extended, or named IP access list number.
- 

Purpose: To filter outgoing route updates to a particular BGP neighbor. Only one distribute list can be used per neighbor. The operation of the output distribute list is identical for both IBGP and EBGP neighbors.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0, support for named access lists was added in Release 11.2, and prefix list support was added in Release 12.0.

### Configuration Example 1: Block a Particular Route

In [Figure 8-9](#), Router B is advertising four network prefixes to Router A. Router B filters the route update to Router A in order to reject the 172.16.2.0 network. Loopbacks are used on Router B to simulate the advertised networks, as shown in the following configuration.





## 8-8 neighbor { ip-address | peer-group-name } ebgp-multihop





## 8-9 neighbor { ip-address | peer-group-name } ebgp-multihop maximum-hop-count

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- maximum-hop-count— Optional parameter with a value of 1 to 255. The default hop count is 255.
- 

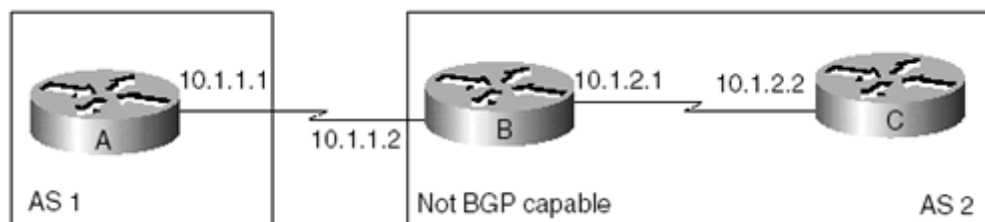
Purpose: EBGP neighbors are typically directly connected. In situations in which EBGP neighbors are not directly connected, the ebgp-multihop option must be used in order to form a neighbor relationship.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Nonconnected EBGP Neighbors

[Figure 8-10](#) shows a situation in which we are trying to establish an EBGP connection between two routers that are not directly connected. BGP uses TCP, and Routers A and C must have routes to each other. The static routes on Routers A and C are necessary in order for BGP to establish the connection. On Router A, the default hop count of 255 is used. On Router C, a maximum hop count of 2 is used. A maximum hop count of 1 is interpreted as directly connected and therefore has no effect.

**Figure 8-10. ebgp-multihop Is Used to Form an EBGP Connection Between Peers That Are Not Directly Connected**



Router A

```
router bgp 1
```





## 8-10 neighbor { ip-address | peer-group-name } filter-list as-path-list-number in

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- as-path-list-number— IP AS path list number.
- 

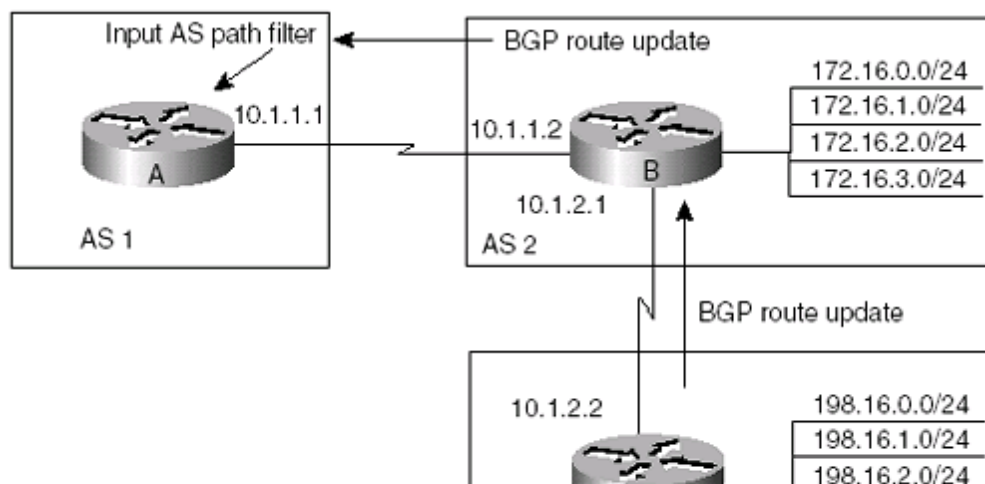
Purpose: To filter incoming route updates from a particular BGP neighbor. Filtering is based on AS path information. Only one filter list can be used per neighbor. The operation of the input filter list is identical for both IBGP and EBGP neighbors.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Block Routes Originating from a Particular AS

In [Figure 8-11](#), Routers B and C are advertising four network prefixes. Router A filters the route update from Router B in order to reject networks originating from AS 3. The last AS listed in the AS path list is the originating AS. Loopbacks are used on Routers B and C to simulate the advertised networks, as shown in the configuration.

**Figure 8-11. Scenario for the Use of the neighbor filter-list in Command**







## 8-11 neighbor { ip-address | peer-group-name } filter-list as-path-list-number out

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- as-path-list-number— IP AS path list number.
- 

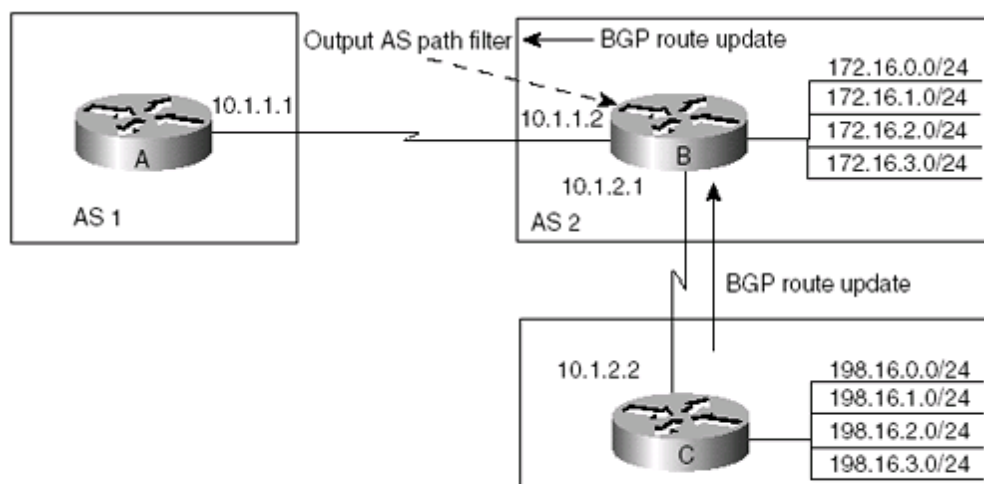
Purpose: To filter outgoing route updates to a particular BGP neighbor. Filtering is based on AS path information. Only one filter list can be used per neighbor. The operation of the output filter list is identical for both IBGP and EBGP neighbors.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Block Routes Originating from a Particular AS

In [Figure 8-12](#), Routers B and C are advertising four network prefixes. Router B filters the route update to Router A in order to reject networks originating from AS 3. The last AS listed in the AS path list is the originating AS. Loopbacks are used on Routers B and C to simulate the advertised networks, as shown in the configuration.

**Figure 8-12. Scenario for the Use of the neighbor filter-list out Command**









## 8-12 neighbor { ip-address | peer-group-name } filter-list as-path-list-number weight weight

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- as-path-list-number— IP AS path list number.
- 
- 
- weight— 1 to 65535. This value is applied to the weight attribute of incoming routes matching the conditions in the AS path filter list.
- 

Purpose: Routes learned from BGP neighbors have the weight attribute set to 0. This form of the filter-list command allows you to set the weight attribute of selected routes received from a particular neighbor. This command applies only to incoming route updates. The operation of the filter list is identical for both IBGP and EBGP neighbors.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0, and the weight keyword was removed in Release 12.1.

### Configuration Example: Set the Weight of Routes Originating from a Particular AS

In [Figure 8-13](#), Routers B and C are advertising four network prefixes. Router A inspects the route update coming from Router B and sets the weight of networks originating from AS 3 to 850. The last AS listed in the AS path list is the originating AS. Loopbacks are used on Routers B and C to simulate the advertised networks, as shown in the configuration.

**Figure 8-13. Scenario for the Use of the neighbor filter-list weight Command**





## 8-13 neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit

**8-14 neighbor** { ip-address | peer-group-name } **maximum-prefix** prefix-limit  
**warning-only**

**8-15 neighbor** { ip-address | peer-group-name } **maximum-prefix** prefix-limit  
threshold-value





## 8-16 neighbor { ip-address | peer-group-name } maximum-prefix prefix-limit threshold-value **warning-only**

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- prefix-limit— 1 to 4294967295.
- 
- 
- threshold-value— 1 to 100 percent. If this isn't explicitly set, the default value is 75 percent.
- 

Purpose: To limit the number of prefixes learned from a specific neighbor. The threshold-value determines the value that causes the router to generate a warning. For example, if the prefix-limit is set to 1000 and the threshold-value is set to 75 percent, the router generates a warning when 751 prefixes are received from the neighbor. When the number of prefixes received from the neighbor exceeds the prefix-limit, the BGP connection between the neighbors is terminated. If the warning-only option is used, the router issues a warning when the prefix limit has been exceeded, but the connection is not terminated.

Cisco IOS Software Release: 11.3

### Configuration Example: Controlling the Maximum Prefixes Learned from a BGP Neighbor

The configuration shown in [Figure 8-14](#) demonstrates the maximum-prefix commands. Router A is configured with a prefix-limit of 8. The nine loopback interfaces on Router B are used to generate the prefixes that are advertised to Router A.

**Figure 8-14. Configuration Used to Demonstrate the maximum-prefix Commands**







## 8-17 neighbor { ip-address | peer-group-name } next-hop-self

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

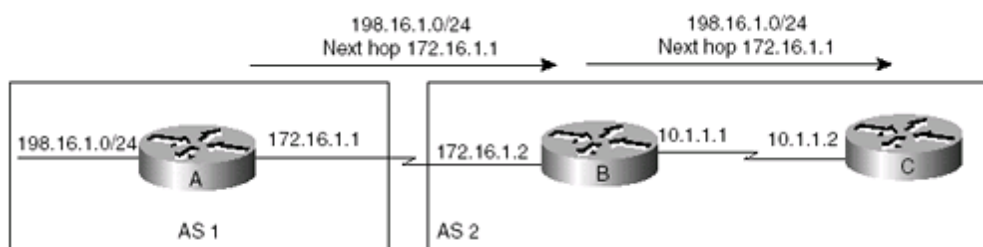
Purpose: When a BGP router learns routes via EBGP, and those routes are advertised to an IBGP neighbor, the next-hop information is sent unchanged. This command allows a BGP router to change the next-hop information that is sent to IBGP peers. The next-hop information is set to the IP address of the interface used to communicate with the neighbor.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Setting Next-Hop Information for Advertised Prefixes

Every prefix that is advertised using BGP contains next-hop information. [Figure 8-15](#) shows the next-hop behavior for EBGP and IBGP. The advertisement for network 198.16.1.0/24 from Router B to Router A contains a next hop of 172.16.1.1. EBGP next-hop information is preserved when the prefix is advertised via IBGP. Router B advertises network 198.16.1.0/24 to Router C, with the next-hop information received from Router A.

**Figure 8-15. Next Hop for EBGP and IBGP Connections**



On a multiaccess network, such as Ethernet or Token Ring, the next-hop behavior is as shown in [Figure 8-16](#). Router B learns about network 198.16.1.0/24 from Router A via IGP. Router B advertises 198.16.1.0/24 to Router C via EBGP with the next-hop information set to the address of Router A. This is done to avoid the extra hop of sending packets destined to network 198.16.1.0/24 to Router B.

**Figure 8-16. Next Hop for Multiaccess Networks**





## 8-18 neighbor { ip-address | peer-group-name } password password

Syntax Description:

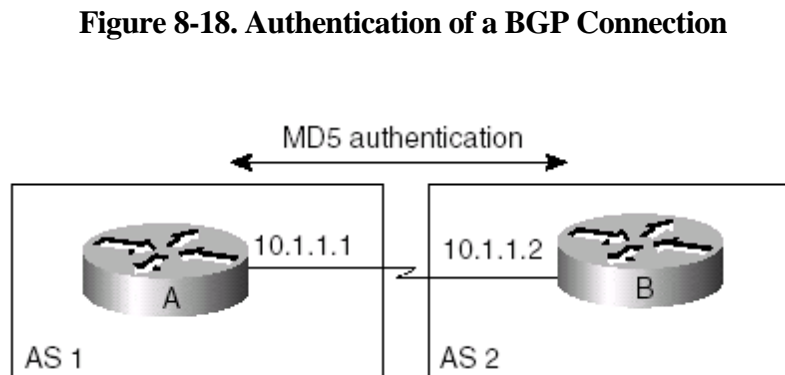
- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- password— Case-sensitive password. The length of the password can be up to 80 characters. The first character of the password cannot be a number. The password can contain any alphanumeric characters, including spaces. For operational reasons, do not use a space after a number.
- 

Purpose: To enable Message Digest 5 (MD5) authentication on a TCP connection between two BGP peers.

Cisco IOS Software Release: 11.0

### Configuration Example: Enabling MD5 Authentication on a TCP Connection Between BGP Peers

The network shown in [Figure 8-18](#) is used to demonstrate password configuration between neighbors.



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```





## 8-19 neighbor peer-group-name peer-group



## 8-20 neighbor ip-address peer-group peer-group-name

Syntax Description:

- 
- peer-group-name— Name of the peer group to create or the name of the peer group to which the neighbor will be added.
- 
- 
- ip-address— IP address of the neighbor to be placed in the peer group with the name peer-group-name.
- 

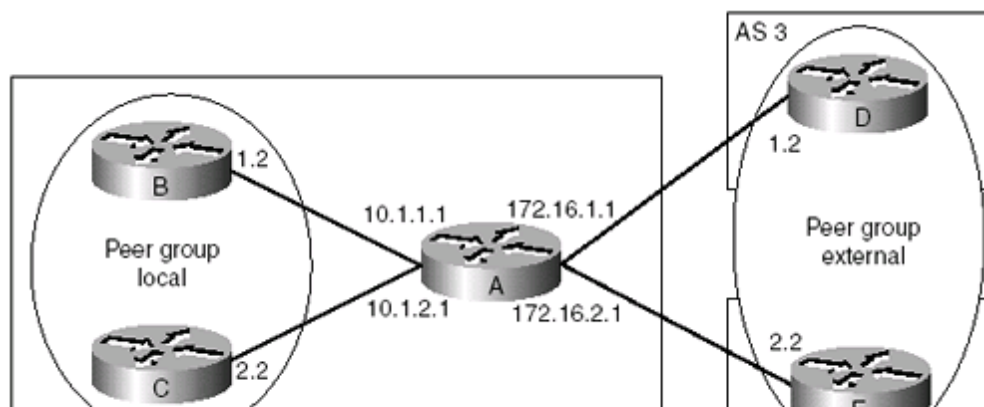
Purpose: Use the neighbor peer-group-name peer-group command to create a BGP peer group. The neighbor ip-address peer-group peer-group-name command adds a neighbor to an existing peer group. Assume that a router has multiple BGP neighbors and that it has identical update policies with those neighbors. The update policy could be configured for each neighbor. In this case, BGP would calculate a separate update for each neighbor even though the updates are identical. If the update policy is applied to a peer group and the neighbors are members of the peer group, the update is calculated once and then sent to all the neighbors in the peer group.

Cisco IOS Software Release: 11.0

### Configuration Example: Creating Peer Groups

Router A in [Figure 8-19](#) has two IBGP peers and two EBGP peers. Assume that Router A has the same update policy for the IBGP peers and the EBGP peers. The IBGP peers could be configured in one peer group and the EBGP peers in another. The first configuration is without using peer groups. The second configuration uses peer groups so that you can compare the syntax.

Figure 8-19. BGP Peer Groups







## 8-21 : neighbor { ip-address | peer-group-name } prefix-list prefix-list-name in

Syntax Description:

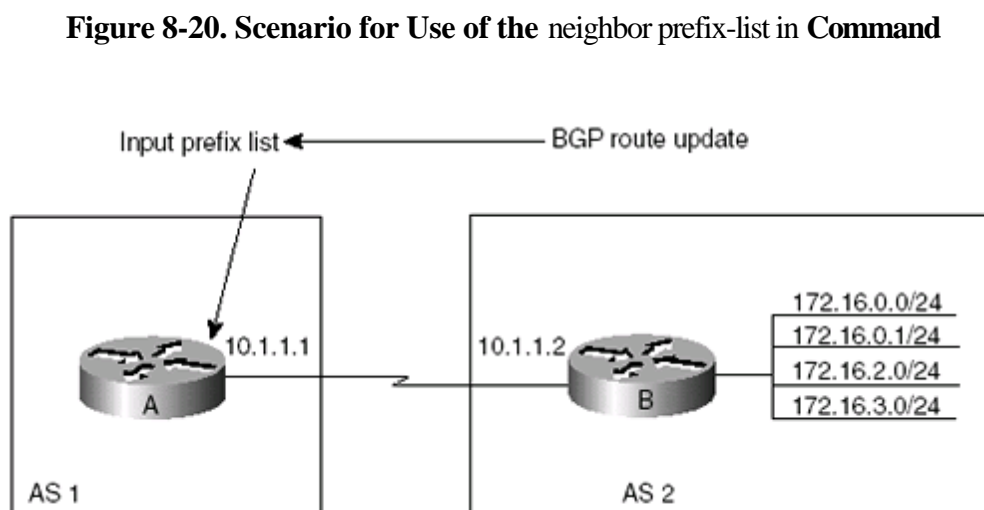
- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- prefix-list-name— Name of the input IP prefix list.
- 

Purpose: To filter incoming route updates from a particular BGP neighbor based on the IP address and mask length. Only one prefix list can be used per neighbor. The operation of the input prefix list is identical for both IBGP and EBGP neighbors. Using a prefix list is an alternative to using an extended IP access list and a distribute list.

Cisco IOS Software Release: 12.0

### Configuration Example 1: Allow an Aggregate Route While Blocking the More-Specific Routes

In [Figure 8-20](#), Router B is advertising four network prefixes and the aggregate of the prefixes to Router A. Router A filters the route update from Router B in order to reject the more-specific routes. Loopbacks are used on Router B to simulate the advertised networks, as shown in the configuration.









## 8-22 neighbor { ip-address | peer-group-name } prefix-list prefix-list-name out

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- prefix-list-name— Name of the output IP prefix list.
- 

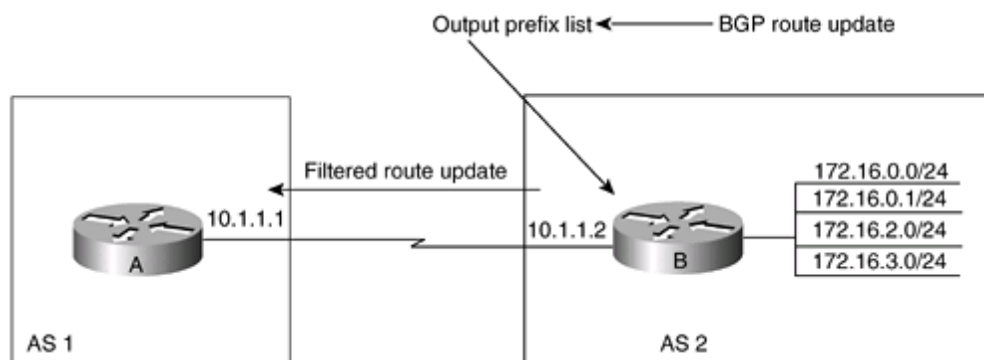
Purpose: To filter outgoing route updates to a particular BGP neighbor based on the IP address and mask length. Only one prefix list can be used per neighbor. The operation of the output prefix list is identical for both IBGP and EBGP neighbors. Using a prefix list is an alternative to using an extended IP access list and a distribution list.

Cisco IOS Software Release: 12.0

### Configuration Example 1: Allow an Aggregate Route While Blocking the More-Specific Routes

In [Figure 8-21](#), Router B is advertising four network prefixes and the aggregate of the prefixes to Router A. Router B filters the route update to Router A in order to reject the more-specific routes. Loopbacks are used on Router B to simulate the advertised networks, as shown in the configuration.

**Figure 8-21. Scenario for Using the neighbor prefix-list out Command**







## 8-23 neighbor { ip-address | peer-group-name } remote-as number

### Syntax Description

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- number— Neighbor's autonomous system number (1 to 65534).
- 

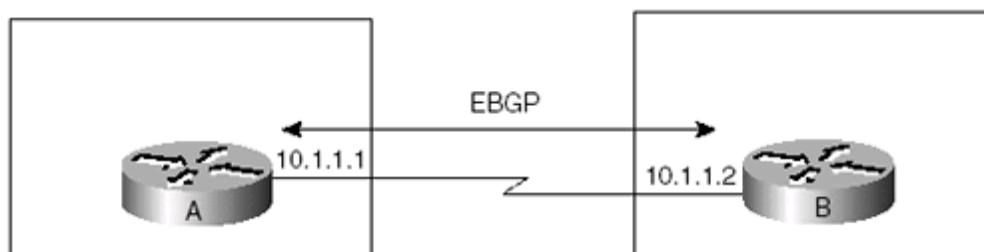
**Purpose:** This form of the neighbor command is the most important. This command is used to configure an internal BGP (IBGP) or external BGP (EBGP) TCP session with another router. Routing information cannot be exchanged without a proper neighbor configuration. When configuring BGP, either in a customer's network or on the CCIE exam, we suggest that you first establish and verify that the neighbor relationships have been established. A common mistake is to attempt to enter a complete BGP configuration, which might include route exchange, route filtering, attribute manipulation, and route redistribution. If the configuration does not produce the intended results, it is usually very difficult to debug due to the configuration complexity. The preferred method is to configure BGP in steps, with the neighbor command being the first step. If neighbors are properly configured, you can continue to satisfy the other requirements.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: EBGP Neighbor

EBGP is used between neighbors in different autonomous systems. Typically, EBGP neighbors are directly connected, as shown in [Figure 8-22](#). If the neighbors are not directly connected, `bgp multihop` can be used (see [sections 8-8](#) and [8-9](#)). For this example, the neighbors are directly connected. The following configuration illustrates the use of the neighbor command to establish an EBGP relationship.

**Figure 8-22. EBGP Neighbor Relationship**







## 8-24 neighbor { ip-address | peer-group-name } remove-private-as

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

Purpose: To remove private autonomous systems in updates to the neighbor or peer group. Private AS numbers are in the range 64512 to 65535. Private AS numbers should not be advertised to the Internet. The following conditions apply when using this command:

- 
- Use only with EBGP peers.
- 
- 
- If the update has only private AS numbers in the AS path, BGP removes them.
- 
- 
- If the AS path includes both private and public AS numbers, BGP doesn't remove the private AS numbers. This situation is considered a configuration error.
- 
- 
- If the AS path contains the AS number of the EBGP neighbor, BGP doesn't remove the private AS number.
- 
- 
- If the AS path contains confederations, BGP removes the private AS numbers only if they come after the confederation portion of the AS path.
- 

Cisco IOS Software Release: 12.0

### Configuration Example: Removing a Private AS Number from Updates to Neighbors or Peer Groups

In [Figure 8-24](#), an ISP is connected to a customer who is using a private AS number. The ISP is connected to another ISP for Internet connectivity. The ISP's AS 1 advertises the private AS number of







## 8-25 neighbor { ip-address | peer-group-name } route-map route-map-name in

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- route-map-name— Name of the route map used for incoming updates from the specified neighbor or peer group.
- 

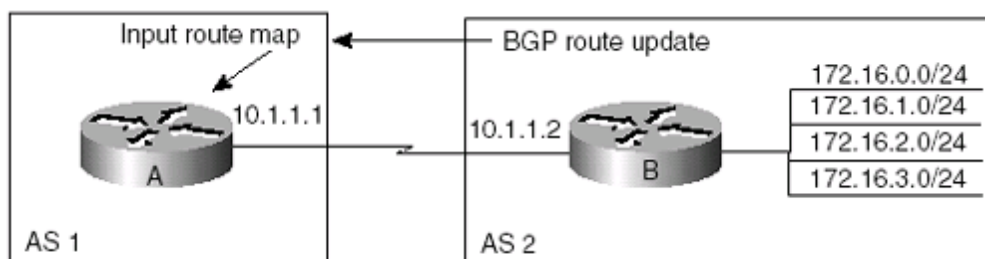
Purpose: A route map is an extremely powerful tool for route filtering and BGP attribute manipulation. [Appendix C](#) contains a complete discussion of route map logic. In this section, we will examine common uses of a route map for route filtering and BGP attribute manipulation.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Basic Route Filter Using an IP Standard Access List

The configuration in [Figure 8-25](#) will be used for each route map example in this section.

**Figure 8-25. Configuration Used to Demonstrate the Use of an Input Route Map**



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```





## 8-26 neighbor { ip-address | peer-group-name } route-map route-map-name out

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- route-map-name— Name of the route map used for outgoing updates to a specified neighbor or peer group.
- 

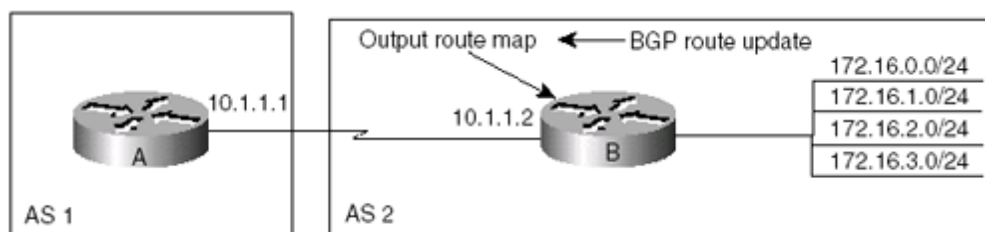
Purpose: A route map is an extremely powerful tool for route filtering and BGP attribute manipulation. [Appendix C](#) contains a complete discussion of route map logic. In this section, we will examine common uses of a route map for route filtering and BGP attribute manipulation.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Basic Route Filter Using an IP Standard Access List

The configuration shown in [Figure 8-26](#) will be used for each route map example in this section.

**Figure 8-26. Configuration Used to Demonstrate the Use of an Output Route Map**



Router A

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```





## 8-27 neighbor { ip-address | peer-group-name } route-reflector-client

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

Purpose: IBGP neighbors do not propagate routing information learned from one IBGP neighbor to another IBGP neighbor. Therefore, if you are running IBGP, every IBGP speaker must have a connection to every other IBGP speaker in the AS. This becomes a scaling problem as the number of IBGP speakers increases. The number of IBGP connections for n speakers is  $(n(n-1))/2$ . [Table 8-1](#) lists the number of connections needed for 2 to 10 IBGP speakers.

Table 8-1. IBGP Connections Needed for a Full Mesh

Number of IBGP Speakers	Number of Connections
2	1
3	3
4	6
5	10
6	15
7	21
8	28
9	36
10	45

A route reflector is one technique to overcome the scaling issue with IBGP. One or more routers serve as a route reflector, and other routers are clients to the route reflector. Route reflectors reflect routes learned from a route reflector client to the other clients. With one route reflector, the number of logical connections needed for n IBGP speakers is n-1.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example 1: Single Route Reflector







## 8-28 neighbor { ip-address | peer-group-name } send-community

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

Purpose: By default, BGP community attributes are not advertised to peers. The neighbor send-community command enables the sending of BGP community attributes to BGP peers. Routing policies can be based on a neighbor address, a peer group name, or AS path information. Situations might arise in which you need to apply policies to routes that do not have any of the previously mentioned attributes in common. A community value is a numerical value or set of values that can be attached to a BGP route. Routing policies can then be applied to routes that contain a particular community value or attribute.

Cisco IOS Software Release: 10.3. The peer-group-name option was added in Release 11.0.

### Configuration Example: NO-EXPORT Community Value

Two well-known community values are NO-EXPORT and NO-ADVERTISE. If a route carries the NO-EXPORT community value, the route is not advertised outside the AS. The behavior of the NO-EXPORT community value is illustrated in [Figure 8-29](#). The NO-ADVERTISE community value prevents a router from advertising the route to any peer, as shown in [Figure 8-30](#). To configure a COMMUNITY attribute, we will use an outbound route map on Router A, as shown in the following configuration. For a complete discussion of route maps, see [Appendix C](#).

Figure 8-29. NO-EXPORT Community

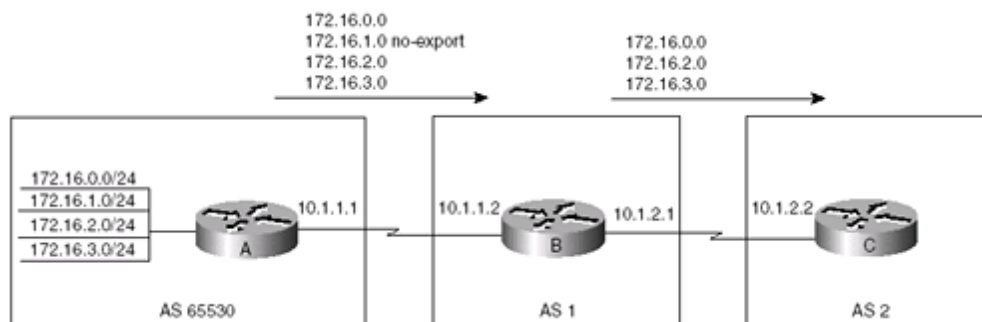


Figure 8-30. NO-ADVERTISE Community





## 8-29 neighbor {ip-address | peer-group-name} shutdown

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

Purpose: Prior to Cisco IOS Software Release 12.0, the only way you could shut down a BGP neighbor was to remove the neighbor configuration using no neighbor ip-address remote-as as-number. This was an inconvenience, because all configuration statements associated with the neighbor were also deleted. The shutdown form of the neighbor command terminates the BGP session without deleting the neighbor configuration commands.

Cisco IOS Software Release: 12.0

### Configuration Example: Administratively Shutting Down a BGP Session

The following configuration administratively shuts down the BGP session. To reenabte the session, use the no form of the command:

```
router bgp 1
  neighbor 10.1.1.2 remote-as 1
  neighbor 10.1.1.2 shutdown
```

### Verification

Use the show ip bgp neighbors command to verify that the neighbor has been shut down:

```
rtrA#show ip bgp neighbors
BGP neighbor is 10.1.1.2, remote AS 1, internal link
```

Administratively shut down

Index 1, Offset 0, Mask 0x2







## 8-30 neighbor { ip-address | peer-group-name } soft-reconfiguration inbound

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 

**Purpose:** If you have a policy configured with a BGP neighbor, such as a route map or distribute list, and you change the policy, the BGP session needs to be cleared in order for the new policy to take effect. When a BGP session is cleared, the cache is invalidated. This has a momentary impact on your routing. The soft-reconfiguration option allows you to change policies without clearing the BGP session. The two forms of soft reconfiguration are inbound and outbound. When you use inbound, soft reconfiguration updates from a neighbor are stored in memory, regardless of the inbound policy. Be aware that using inbound soft reconfiguration uses more memory than not using inbound soft reconfiguration. Outbound soft reconfiguration does not require additional memory and is always enabled.

Cisco IOS Software Release: 11.2

### Configuration Example: Setting Inbound Soft Reconfiguration with a Specific Neighbor

To configure inbound soft reconfiguration with a specific neighbor, use the following configuration as a guide:

```
router bgp 2
  neighbor 10.1.1.2 remote-as 1
  neighbor 10.1.1.2 soft-reconfiguration inbound
```

### Verification

Use the show ip bgp neighbors command to verify that soft reconfiguration has been enabled:

```
router#
```

```
BGP neighbor is 10.1.1.2, remote AS 1, external link
```

```
Index 1, Offset 0, Mask 0x2
```





## 8-31 neighbor { ip-address | peer-group-name } timers keepalive holdtime

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- keepalive— 1 to 4,294,967,295 seconds.
- 
- 
- holdtime— 1 to 4,294,967,295 seconds.
- 

**Purpose:** Keepalive and holdtime are common among IP routing protocols. The keepalive time indicates how often a router sends a keepalive message to a neighbor to inform the neighbor that the router is still alive and well. The holdtime is used as a deathwatch. If a keepalive message is not received within the holdtime, the neighbor is declared dead, and the session is terminated. The default value for the keepalive time is 60 seconds. The holdtime is 3 times the keepalive time, or 180 seconds. Generally, these values do not need to be changed. If you do change them, it is a good rule to make the holdtime equal to 3 times whatever keepalive value you use. Of course, the holdtime should always be greater than the keepalive time. A good practice to follow is to configure the same keepalive and holdtimes on both sides of the link.

Cisco IOS Software Release: 12.0

### Configuration Example: Changing the Keepalive and Holdtime Values

Change the default settings for keepalive and holdtime to 50 and 150 seconds, respectively:

```
Router A
```

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```

```
neighbor 10.1.1.2 timers 50 150
```

```
no auto-summary
```





## 8-32 neighbor { ip-address | peer-group-name } **unsuppress-map** route-map-name

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- route-map-name— Name of the route map used to select routes to be unsuppressed.
- 

Purpose: When the aggregate-address command is used with the summary-only option, the more-specific routes of the aggregate are suppressed (see [section 1-7](#)). The aggregate-address summary-only command suppresses the more-specific routes to all neighbors. You can use an unsuppress map to selectively leak more-specific routes to a particular neighbor.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Selectively Advertising Routes with an Unsuppress Map

In [Figure 8-31](#), Router A is advertising four prefixes to Router B. Router B aggregates these prefixes and advertises only the summary while suppressing the four specific routes. This will be used as the initial configuration so that we can inspect the BGP tables for Routers B and C. The unsuppress map is then added to Router B's configuration.

**Figure 8-31. Globally Suppressed Routes Can Be Selectively Unsuppressed on a Per-Neighbor Basis**









## 8-33 neighbor { ip-address | peer-group-name } update-source interface-name

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- 
- interface-name— Any physical or logical router interface. Usually, a loopback interface is used.
- 

Purpose: An IBGP connection can occur as long as there is a TCP/IP path between the routers. If multiple paths exist between the IBGP routers, using a loopback interface as the neighbor's address can add stability to the network. Using loopback interfaces with EBGP speakers is not necessary, because EBGP neighbors are usually directly connected.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Using a Loopback Interface for Network Stability

In [Figure 8-32](#), we want to establish an IBGP session between Routers A and C. From Router A's point of view, we can use one of two addresses in the neighbor command, 172.17.1.2 or 10.1.2.2. From Router C's perspective, either address 172.17.1.1 or 10.1.1.1 can be used to form a neighbor relationship with Router A. The following configurations show these options:

**Figure 8-32. When Multiple Paths Exist Between IBGP Peers, Using a Loopback Interface as the Source Adds Stability to the Design**







## 8-34 neighbor { ip-address | peer-group-name } version version-number

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- version-number— 2, 3, or 4.
- 

Purpose: Currently, the default BGP version number is 4. Cisco's implementation of BGP begins using version 4 and negotiates down to version 2. If you do not want to form a neighbor connection with a router that is not running version 4, you can lock down the version number using this command.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Locking Down the Neighbor BGP Version

The following configuration locks down the version number with one neighbor to 4 and leaves the version number in auto-negotiate for the other neighbors:

```
Router A
```

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```

```
neighbor 10.1.1.2 version 4
```

```
neighbor 10.2.1.2 remote-as 3
```

```
neighbor 10.3.1.2 remote-as 4
```

### Verification

Verify the version number by using the show ip bgp neighbors command:





## 8-35 neighbor { ip-address | peer-group-name } weight default-weight

Syntax Description:

- 
- ip-address— Neighbor's IP address.
- 
- peer-group-name— Name of the peer group. See [section 8-19](#).
- 
- default-weight— 1 to 65535.
- 

Purpose: The weight attribute has meaning only to the local router. By default, advertisements received from BGP neighbors have their weight attribute set to 0. This command sets all routes learned from a particular neighbor to the default weight used in the command.

Cisco IOS Software Release: 10.0. Peer group support was added in Release 11.0.

### Configuration Example: Setting the Route Weight for the Local Router

The following configuration sets the weight of all routes learned from neighbor 10.1.1.2 to 451:

Router A

```
interface Serial0
```

```
ip address 10.1.1.1 255.255.255.252
```

```
!
```

```
router bgp 1
```

```
neighbor 10.1.1.2 remote-as 2
```

```
neighbor 10.1.1.2 weight 451
```

---

Router B

```
interface Loopback0
```

```
ip address 172.16.0.1 255.255.255.0
```





# Chapter 9. Route Advertisement

- [Section 9-1. network](#) ip-address
- 
- [Section 9-2. network](#) ip-address [mask](#) network-mask
- 
- [Section 9-3. network](#) ip-address [backdoor](#)
- 
- [Section 9-4. network](#) ip-address [mask](#) network-mask [backdoor](#)
- 
- [Section 9-5. network](#) ip-address [route-map](#) route-map-name
- 
- [Section 9-6. network](#) ip-address [mask](#) network-mask [route-map](#) route-map-name
- 
- [Section 9-7. network](#) ip-address [weight](#) weight
- 
- [Section 9-8. network](#) ip-address [mask](#) network-mask [weight](#) weight
-

# 9-1 network ip-address



## 9-2 network ip-address mask network-mask

### Syntax Description:

- 
- ip-address— Network to advertise to BGP peers.
- 
- network-mask— Optional parameter used to advertise nonclassful network prefixes.
- 

Defaults: None

Limitations: Up to 200 instances of the network command may be used in the configuration. For Cisco IOS Software Release 12.0 and later, this restriction has been removed.

Purpose: Interior Gateway Protocols such as RIP and OSPF use the network command to determine on which interfaces the protocol will be active. The BGP neighbor command is used to determine which interfaces will run BGP. The BGP network command is used to determine the networks that will be advertised to BGP neighbors. In order for a network to be advertised by BGP, it must be known to the originating router. Routes learned via EBGP are automatically advertised to other EBGP neighbors. A known network is one that is directly connected, static, or learned through a dynamic routing protocol. The first form of the network command requires a classful IP address. A classful address is either Class A with an 8-bit subnet mask, Class B with a 16-bit subnet mask, or Class C with a 24-bit subnet mask. The second form can be used with either a classful or classless prefix.

Cisco IOS Software Release: 10.0

### Configuration Example 1: Directly Connected Networks

[Figure 9-1](#) illustrates a basic scenario for the use of the network command. Router A has two directly connected networks that are advertised to router B via BGP.

**Figure 9-1. Basic Use of the network Command**





## 9-3 network ip-address **backdoor**





## 9-4 network ip-address mask network-mask backdoor

Syntax Description:

- 
- ip-address— Network to advertise to BGP peers.
- 
- 
- network-mask— Optional parameter used to advertise nonclassful network prefixes.
- 

Defaults: None

Limitations: Up to 200 instances of the network command may be used in the configuration. For Cisco IOS Software Release 12.0 and later, this restriction has been removed.

Purpose: When a router is running more than one IP routing protocol, the possibility exists that a particular route might be learned by two or more protocols. Because different IP routing protocols calculate the cost to a route using different metrics, the protocol cost cannot be used to select the best path. When a route is known by more than one IP routing protocol, Cisco routers use the administrative distance to select the best path, with the lowest administrative distance being preferred. EBGp routes have an administrative distance of 20, and IGP's have a higher administrative distance:

- 
- EBGp— 20
- 
- 
- EIGRP— 90
- 
- 
- IGRP— 100
- 
- 
- OSPF— 110
- 
- 
- RIP— 120
- 
-



**9-5 network** ip-address **route-map** route-map-name

## 9-6 network ip-address **mask** network-mask **route-map** route-map-name

Purpose: In theory, these commands allow you to modify a network's BGP attributes. In our experience, they are too buggy and should not be used. See [sections 8-25](#) and [8-26](#) if you need to modify the BGP attributes of received or transmitted network advertisements.

## 9-7 network ip-address **weight** weight

## 9-8 network ip-address **mask** network-mask **weight** weight

Obsolete: These commands don't work and are considered obsolete. They will be removed from future versions of Cisco IOS Software. They are included here because they exist in current Cisco IOS Software Releases, but they do nothing. See [sections8-25](#) or [8-35](#) if you need to modify the weight of received network advertisements.

# Chapter 10. Route Redistribution

- [Section 10-1. redistribute](#) protocol
- 
- [Section 10-2. redistribute](#) protocol [metric](#) metric
- 
- [Section 10-3. redistribute](#) protocol [route-map](#) route-map-name
- 
- [Section 10-4. redistribute](#) protocol [route-map](#) route-map-name [metric](#) metric
- 
- [Section 10-5. redistribute](#) protocol [weight](#) weight
-





## 10-1 redistribute protocol

Syntax Description:

- 
- protocol— Routes learned via protocol will be redistributed into BGP.
- 

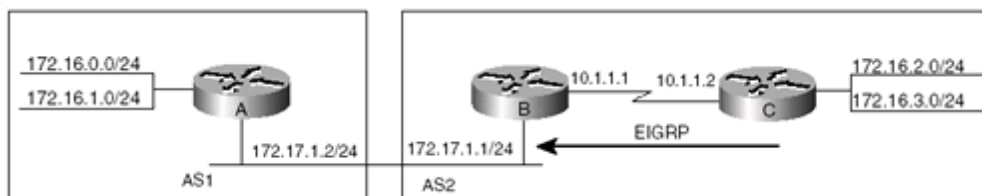
Purpose: To redistribute routes into BGP that have been learned via a routing protocol other than BGP. The metric of the non-BGP-learned routes is transferred to the metric or multi-exit discriminator (MED) of the new BGP route. Routes can be redistributed from connected, dvmrp, egp, eigrp, igrp, isis, iso-igrp, mobile, odr, ospf, rip, and static.

Cisco IOS Software Release: 10.0

### Configuration Example: Redistributing Connected, Static, and EIGRP Learned Routes into BGP

In [Figure 10-1](#), Router C is advertising 172.16.2.0/24 and 172.16.3.0/24 to Router B via EIGRP. Routers A and B have an EBGP relationship. This example redistributes EIGRP and static into BGP on Router B and redistributes connected on Router A.

**Figure 10-1. Redistributing Routes into BGP**



Router A

```
interface loopback 0
```

```
ip address 172.16.0.1 255.255.255.0
```

```
!
```

```
interface loopback 1
```

```
ip address 172.16.1.1 255.255.255.0
```

```
!
```





## 10-2 redistribute protocol metric metric

Syntax Description:

- 
- protocol— Routes learned via protocol will be redistributed into BGP.
- 
- 
- metric— Metric to assign to the redistributed routes. The value is in the range of 0 to 4294967295.
- 

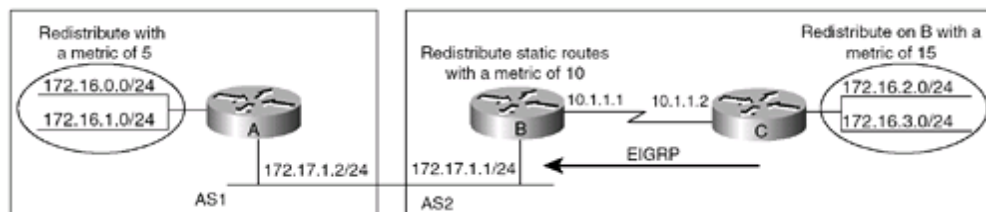
Purpose: To redistribute routes into BGP that have been learned via a routing protocol other than BGP. The metric value is assigned to the metric or multi-exit discriminator (MED) of the new BGP route. Routes can be redistributed from connected, dvmrp, egp, eigrp, igrp, isis, iso-igrp, mobile, odr, ospf, rip, and static.

Cisco IOS Software Release: 10.0

### Configuration Example: Redistributing Connected, Static, and EIGRP Learned Routes into BGP

In [Figure 10-2](#), Router C is advertising 172.16.2.0/24 and 172.16.3.0/24 to Router B via EIGRP. Routers A and B have an EBGP relationship. This example redistributes EIGRP and static into BGP on Router B and redistributes connected on Router A. The static routes are assigned a metric of 10, the EIGRP routes a metric of 15, and the connected routes a metric of 5.

**Figure 10-2. Redistributing Routes into BGP**



Router A

```
interface loopback 0
```

```
ip address 172.16.0.1 255.255.255.0
```

```
!
```

```
interface loopback 1
```



## **10-3 redistribute** protocol **route-map** route-map-name



## 10-4 redistribute protocol route-map route-map-name metric metric

Syntax Description:

- 
- protocol— Routes learned via protocol will be redistributed into BGP.
- 
- 
- route-map-name— Name of the route map used to control which routes will be redistributed into BGP. The route map can also be used to modify the BGP attributes of the redistributed routes.
- 
- 
- metric— Metric to assign to the redistributed routes. The value is in the range of 0 to 4294967295.
- 

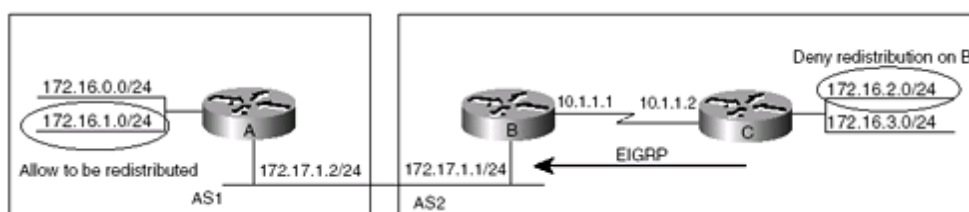
Purpose: To redistribute routes into BGP that have been learned via a routing protocol other than BGP. The metric of the non-BGP-learned routes is transferred to the metric or multi-exit discriminator (MED) of the new BGP route if the metric option is not used. If the metric option is used, the assigned metric will be applied to all routes redistributed from the protocol. A route map can be used for each redistributed protocol to control which routes are redistributed. A route map can also be used to modify the BGP attributes of the redistributed routes. Routes can be redistributed from connected, dvmrp, egp, eigrp, igrp, isis, iso-igrp, mobile, odr, ospf, rip, and static.

Cisco IOS Software Release: 10.0

### Configuration Example: Selectively Redistributing Connected, Static, and EIGRP Learned Routes into BGP

In [Figure 10-3](#), Router C is advertising 172.16.2.0/24 and 172.16.3.0/24 to Router B via EIGRP. Routers A and B have an EBGP relationship. This example redistributes EIGRP and static into BGP on Router B and redistributes connected on Router A. On Router B, you want to block the redistribution of the EIGRP route 172.16.2.0. On Router A, you want to allow only the redistribution of network 172.16.1.0/24. Another route map is used on Router B to set the weight of the redistributed static route to 88.

Figure 10-3. Selectively Redistributing Routes into BGP







## 10-5 redistribute protocol weight weight

Syntax Description:

- 
- protocol— Routes learned via protocol will be redistributed into BGP.
- 
- 
- weight— Weight assigned to the redistributed routes. The value is in the range of 0 to 65535.
- 

Obsolete: This command is considered obsolete. It will be removed from future versions of the IOS. This command is included here because it exists in current IOS versions. Use a route-map (see [section 10-4](#)) if you need to set the weight of redistributed routes.

# Chapter 11. Address Summarization

- [Section 11-1. summary-address](#)
-

## 11-1 summary-address

Purpose: The summary-address command works only with OSPF and IS-IS. Even though this command appears as a BGP router configuration option, do not use this command when configuring BGP. Use the aggregate-address commands described in [Chapter 1](#), "Route Aggregation."

# Chapter 12. Synchronization

- [Section 12-1. synchronization](#)
-



# 12-1 synchronization

Syntax Description:

This command has no arguments.

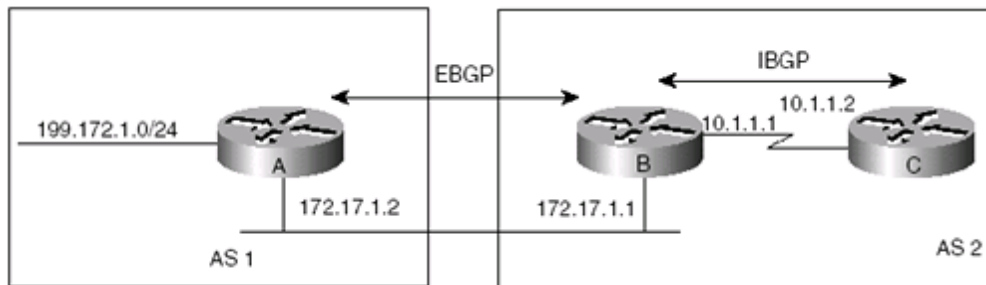
Purpose: IBGP learned routes are not installed in the IP routing table if the route has not been learned by an IGP or if the advertised next hop is inaccessible. This is called synchronization.

IOS Release: 10.0

## Configuration Example: BGP Synchronization

In [Figure 12-1](#), BGP synchronization is enabled by default. Router A is advertising network 199.172.1.0/24 to Router B via EBGP, and Router B is advertising 199.172.1.0/24 to Router C via IBGP.

**Figure 12-1. Synchronization Applies to IBGP Routes**



Router A

```
interface Loopback0
```

```
ip address 199.172.1.1 255.255.255.0
```

```
!
```

```
router bgp 1
```

```
neighbor 172.17.1.1 remote-as 2
```

```
!
```

---

Router B





## Chapter 13. BGP Timers

- [Section 13-1. timers bgp](#) keepalive holdtime
-



## 13-1 timers `bgp` `keepalive` `holdtime`

Syntax Description:

- 
- `keepalive`— 1 to 4294967295 seconds.
- 
- 
- `holdtime`— 1 to 4294967295 seconds.
- 

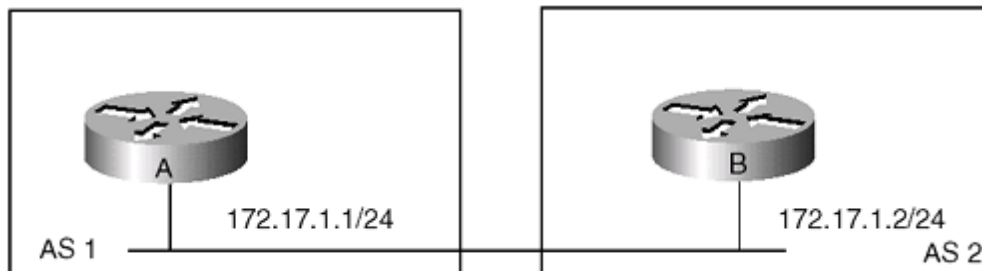
**Purpose:** To globally set the `keepalive` and `holdtime` values for all neighbors. `Keepalive` and `holdtime` are common among IP routing protocols. The `keepalive` time indicates how often a router sends a `keepalive` message to a neighbor to inform the neighbor that the router is still alive and well. The `holdtime` is used as a deathwatch. If a `keepalive` message is not received within the `holdtime`, the neighbor is declared dead, and the session is terminated. The default value for the `keepalive` time is 60 seconds. The `holdtime` is three times the `keepalive` time, or 180 seconds. Generally, these values do not need to be changed. If you do change them, it is a good rule to make the `holdtime` equal to three times whatever `keepalive` value you use. Of course, the `holdtime` should always be greater than the `keepalive` time. When two BGP speakers establish a connection, the smaller of the advertised `keepalive` and `holdtime` values will be used. For example, if one neighbor advertises a `keepalive` value of 70 and a `holdtime` of 210 and the other neighbor advertises a `keepalive` value of 72 and a `holdtime` of 205, the `keepalive` value will be 70 and the `holdtime` will be 205. If the timer values are set using the `neighbor timers` command (see [section 8-31](#)), these values will override the global values only for that neighbor.

IOS Release: 10.0

### Configuration Example: BGP Timers

In this example, we will examine the result of various settings of `keepalive` and `holdtimers`. [Figure 13-1](#) illustrates the basic setup for these examples.

**Figure 13-1. Network Setup for BGP Timer Examples**







# Chapter 14. BGP show Commands

- [Section 14-1. show ip bgp](#)
- 
- [Section 14-2. show ip bgp | begin line](#)
- 
- [Section 14-3. show ip bgp | exclude line](#)
- 
- [Section 14-4. show ip bgp | include line](#)
- 
- [Section 14-5. show ip bgp prefix](#)
- 
- [Section 14-6. show ip bgp prefix mask](#)
- 
- [Section 14-7. show ip bgp prefix/mask-length](#)
- 
- [Section 14-8. show ip bgp prefix mask longer-prefixes](#)
- 
- [Section 14-9. show ip bgp prefix/mask-length longer-prefixes](#)
- 
- [Section 14-10. show ip bgp prefix | begin line](#)
- 
- [Section 14-11. show ip bgp prefix | exclude line](#)
- 
- [Section 14-12. show ip bgp prefix | include line](#)
- 
- [Section 14-13. show ip bgp prefix mask | begin line](#)
- 
- [Section 14-14. show ip bgp prefix mask | exclude line](#)
- 
- [Section 14-15. show ip bgp prefix mask | include line](#)
- 
- [Section 14-16. show ip bgp prefix/mask-length | begin line](#)



## 14-1 show ip bgp



## 14-2 show ip bgp | begin line

## 14-3 show ip bgp | exclude line



## 14-4 show ip bgp | include line

Syntax Description:

- 
- line— Regular expression. See [Appendix B](#), "Regular Expressions."
- 

Purpose: To display the contents of the local BGP routing table.

Cisco IOS Software Release: 10.0

### Example: Display the BGP Routing Table

The following is some sample output from the show ip bgp command:

```
rtrA#show ip bgp
```

```
BGP table version is 9, local router ID is 200.1.4.1
```

```
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
*> 199.172.1.0	172.17.1.2	0		0	1 i
*> 199.172.2.0	172.17.1.2	0		0	1 i
*> 199.172.3.0	172.17.1.2	0		0	1 i
*> 199.172.4.0	172.17.1.2	0		0	1 i
*> 200.1.1.0	0.0.0.0	0		32768	i
*s 200.1.2.0	0.0.0.0	0		32768	i
*> 200.1.2.0/23	0.0.0.0			32768	i
*s 200.1.3.0	0.0.0.0	0		32768	i
*> 200.1.4.0	0.0.0.0	0		32768	i
*>i156.26.1.0	10.1.1.2	0		0	i



## 14-5 show ip bgp prefix

Syntax Description:

- 
- prefix— IP address of the prefix to display.
- 

Purpose: To display the contents of the local BGP routing table for a specific prefix.

Cisco IOS Software Release: 10.0

### Example: Display a Specific Prefix from the BGP Routing Table

The following is some sample output from the show ip bgp prefix command:

```
rtrA#show ip bgp 199.172.1.0
BGP routing table entry for 199.172.1.0/24, version 6
Paths: (1 available, best #1)
  Advertised to non peer-group peers:
    172.17.1.1
Local
  0.0.0.0 from 0.0.0.0 (199.172.15.1)
    Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local,
best, ref 2
```

## 14-6 show ip bgp prefix mask





## 14-7 show ip bgp prefix/mask-length

### Syntax Description:

- 
- prefix— IP address of the prefix to display.
- 
- 
- mask— Display prefix with a specific network mask.
- 
- 
- mask-length— Bit length of the mask.
- 

Purpose: To display the contents of the local BGP routing table for a specific prefix with a specific mask.

Cisco IOS Software Release: 10.0

### Example: Display a Specific Prefix from the BGP Routing Table with the Specified Mask

The following is some sample output from the show ip bgp prefix mask command. First, list the entire contents of the BGP routing table:

```
rtrA#show ip bgp
```

```
BGP table version is 90, local router ID is 199.172.15.1
```

```
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
*> 199.172.1.0	0.0.0.0	0		32768	i
s> 199.172.2.0	0.0.0.0	0		32768	i
*> 199.172.2.0/23	0.0.0.0			32768	i
s> 199.172.3.0	0.0.0.0	0		32768	i
*> 199.172.4.0	0.0.0.0	0		32768	i



## 14-8 show ip bgp prefix mask longer-prefixes



## 14-9 show ip bgp prefix/mask-length longer-prefixes

Syntax Description:

- 
- prefix— IP address of the prefix to display.
- 
- 
- mask— Display prefix with a specific network mask.
- 
- 
- mask-length— Bit length of the mask.
- 

Purpose: To display the contents of the local BGP routing table for a specific prefix with a specific mask and any prefixes having a longer mask than the one specified.

Cisco IOS Software Release: 10.0

### Example: Display a Specific Prefix from the BGP Routing Table Having a Mask Greater Than or Equal to the Mask Specified in the Command

The following is some sample output from the show ip bgp prefix mask longer-prefixes command. First, list the entire BGP routing table:

```
rtrA#show ip bgp
```

```
BGP table version is 90, local router ID is 199.172.15.1
```

```
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
*> 199.172.1.0	0.0.0.0	0		32768	i
s> 199.172.2.0	0.0.0.0	0		32768	i
*> 199.172.2.0/23	0.0.0.0			32768	i
s> 199.172.3.0	0.0.0.0	0		32768	i
*> 199.172.4.0	0.0.0.0	0		32768	i



## 14-10 show ip bgp prefix | begin line

## 14-11 show ip bgp prefix | exclude line



## 14-12 show ip bgp prefix | include line

**14-13** show ip bgp prefix mask | **begin** line

## 14-14 show ip bgp prefix mask | exclude line

**14-15** show ip bgp prefix mask | **include** line

**14-16** `show ip bgp prefix/mask-length | begin` line

**14-17** `show ip bgp prefix/mask-length | exclude line`

**14-18** `show ip bgp prefix/mask-length | include line`

**14-19** `show ip bgp` prefix mask **longer-prefixes** | **begin** line



**14-20** `show ip bgp prefix mask longer-prefixes | exclude line`

**14-21 show ip bgp** prefix mask **longer-prefixes | include** line

**14-22** `show ip bgp prefix/mask-length longer-prefixes | begin` line

**14-23** `show ip bgp prefix/mask-length longer-prefixes | exclude line`



## 14-24 show ip bgp prefix/mask-length longer-prefixes | include line

Syntax Description:

- 
- prefix— IP address of the prefix to display.
- 
- 
- mask— Display prefix with a specific network mask.
- 
- 
- line— Regular expression. See [Appendix B](#).
- 
- 
- mask-length— Bit length of the mask.
- 

Purpose: To display the contents of the local BGP routing table for a specific prefix or a specific prefix/mask using output modifiers.

Cisco IOS Software Release: 10.0

### Example: Display a Specific Prefix from the BGP Routing Table Using Output Modifiers

The following is some sample output from the show ip bgp prefix mask longer-prefixes | include line command. First, list the entire BGP routing table:

```
rtrA#show ip bgp
```

```
BGP table version is 90, local router ID is 199.172.15.1
```

```
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
*>	199.172.1.0	0.0.0.0	0		32768	i
s>	199.172.2.0	0.0.0.0	0		32768	i



## 14-25 show ip bgp cidr-only



## 14-26 show ip bgp cidr-only | begin line

## 14-27 show ip bgp cidr-only | exclude line



## 14-28 show ip bgp cidr-only | include line

Syntax Description:

- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display the contents of the local BGP routing table for prefixes having a nonnatural mask. A natural mask is 8 bits for a Class A network, 16 bits for Class B, and 24 bits for Class C.

Cisco IOS Software Release: 10.0

### Example: Display BGP Prefixes Having a Nonnatural Mask

First, list the entire BGP routing table:

```
rtrA#show ip bgp
```

```
BGP table version is 90, local router ID is 199.172.15.1
```

```
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
*> 199.172.1.0	0.0.0.0	0		32768	i
s> 199.172.2.0	0.0.0.0	0		32768	i
*> 199.172.2.0/23	0.0.0.0			32768	i
s> 199.172.3.0	0.0.0.0	0		32768	i
*> 199.172.4.0	0.0.0.0	0		32768	i
*> 200.1.1.0	172.17.1.1	0		0	2 i
*> 200.1.2.0	172.17.1.1	0		0	2 i
*> 200.1.3.0	172.17.1.1	0		0	2 i
*> 200.1.4.0	172.17.1.1	0		0	2 i



## 14-29 show ip bgp community community-number(s)

## 14-30 `show ip bgp community` community-number(s) **exact-match**

**14-31 show ip bgp community** community-number(s) | **begin** line



**14-32 show ip bgp community** community-number(s) | **exclude** line

**14-33 show ip bgp community** community-number(s) | **include** line

**14-34 show ip bgp community** `community-number(s)` | **begin** line  
**exact-match**

**14-35 show ip bgp community** community-number(s) | **exclude** line  
**exact-match**

## 14-36 show ip bgp community community-number(s) | include line exact-match

Syntax Description:

- 
- community-number(s)— One or more BGP community numbers. Community numbers are in the range of 1 to 4294967295. Well-known community keywords that may be used are local-as, no-advertise, and no-export (see [section 8-28](#)).
- 
- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display the contents of the local BGP routing table for prefixes matching the supplied community number(s).

Cisco IOS Software Release: 10.3. The local-as keyword was added in Cisco IOS Software Release 12.0.

### Example: Display BGP Prefixes Belonging to a Specific Community

Routes belonging to a specific community can be listed using the show ip bgp community command followed by the community value of interest.

```
rtrA#show ip bgp community no-export
```

```
BGP table version is 41, local router ID is 10.1.1.2
```

```
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
*> 172.16.1.0/24	10.1.1.1	0		0	65530 i



## 14-37 **show ip bgp community-list** community-list-number

**14-38 show ip bgp community-list** community-list-number | **begin** line



**14-39 show ip bgp community-list** `community-list-number` | **exclude** line

**14-40 show ip bgp community-list** `community-list-number` | **include** line

## 14-41 **show ip bgp community-list** community-list-number **exact-match**

**14-42 show ip bgp community-list** `community-list-number` **exact-match** |  
**begin** `line`

**14-43 show ip bgp community-list** `community-list-number` **exact-match** |  
**exclude** `line`



## 14-44 show ip bgp community-list community-list-number exact-match | include line

Syntax Description:

- 
- community-list-number— IP community list number. Valid list numbers are 1 to 199.
- 
- line— Regular expression. See [Appendix B](#), "Regular Expressions."
- 

Purpose: To display the contents of the local BGP routing table for prefixes matching the community listed in the community list.

Cisco IOS Software Release: 10.3

### Example: Display BGP Prefixes Belonging to a Specific Community Using a Community List

An IP community list can be used to list routes belonging to a certain community or list of communities using the following configuration. This example lists only those routes with a community value of no-export.

```
ip community-list 1 permit no-export
```

```
rtrA#show ip bgp community 1
```

```
BGP table version is 41, local router ID is 10.1.1.2
```

```
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
*> 172.16.1.0/24	10.1.1.1	0		0	65530 I





## 14-45 show ip bgp dampened-paths

## 14-46 show ip bgp dampened-paths | begin line

## 14-47 show ip bgp dampened-paths | exclude line

## 14-48 show ip bgp dampened-paths | include line

Syntax Description:

- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display the contents of the local BGP routing table for dampened prefixes (see [sections 3-9](#) through [3-12](#)).

Cisco IOS Software Release: 11.0

### Example: Display BGP Dampened Prefixes

In order to display only those routes that have been dampened, use the show ip bgp dampened-paths command. BGP dampening must be enabled before the router will maintain this information in memory.

```
rtrA#show ip bgp dampened-paths
```

```
BGP table version is 7, local router ID is 10.1.1.2
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	From	Reuse	Path
*d 172.16.2.0/24	10.1.1.1	00:19:30	1 i

## 14-49 **show ip bgp filter-list** as-path-access-list

**14-50 show ip bgp filter-list** as-path-access-list | **begin** line

## 14-51 **show ip bgp filter-list** as-path-access-list | **exclude** line





## 14-52 show ip bgp filter-list as-path-access-list | include line

Syntax Description:

- 
- as-path-access-list— AS path access list number. The range of valid numbers is 1 to 199.
- 
- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display the contents of the local BGP routing table for prefixes matching the specified AS path access list.

Cisco IOS Software Release: 10.0

### Example: Displaying BGP Dampened Prefixes to Match Specified AS Path Access List

The AS path attribute of a BGP route can be used to determine the routes that are displayed when using the show ip bgp command. For this example, we want to use an access list to list only those routes that originated in AS 2. Only routes having the AS value(s) permitted in the access list will be displayed.

```
ip as-path access-list 1 permit ^2$
```

```
rtrA#show ip bgp filter-list 1
```

```
BGP table version is 90, local router ID is 199.172.15.1
```

```
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
*> 200.1.1.0	172.17.1.1	0		0 2	i
*> 200.1.2.0	172.17.1.1	0		0 2	i
*> 200.1.3.0	172.17.1.1	0		0 2	i
*> 200.1.4.0	172.17.1.1	0		0 2	I



## 14-53 show ip bgp flap-statistics

## 14-54 show ip bgp flap-statistics prefix/mask-length

## 14-55 show ip bgp flap-statistics prefix/mask-length longer-prefixes

**14-56 show ip bgp flap-statistics** prefix/mask-length **longer-prefixes** |  
**begin** regular-expression

**14-57 show ip bgp flap-statistics** prefix/mask-length **longer-prefixes** |  
**exclude** regular-expression

**14-58 show ip bgp flap-statistics** prefix/mask-length **longer-prefixes** |  
**include** regular-expression



**14-59** `show ip bgp flap-statistics` prefix/mask-length | **begin** regular-expression

**14-60 show ip bgp flap-statistics** prefix/mask-length | **exclude** regular-expression

**14-61 show ip bgp flap-statistics** prefix/mask-length | **include** regular-expression

## 14-62 show ip bgp flap-statistics prefix

## 14-63 `show ip bgp flap-statistics` `prefix` | `begin` `regular-expression`

## 14-64 `show ip bgp flap-statistics` `prefix` | `exclude` `regular-expression`

**14-65** `show ip bgp flap-statistics` `prefix` | **include** `regular-expression`

## 14-66 show ip bgp flap-statistics prefix mask



**14-67 show ip bgp flap-statistics** prefix mask | **begin** regular-expression

**14-68 show ip bgp flap-statistics** prefix mask | **exclude** regular-expression

**14-69** `show ip bgp flap-statistics` prefix mask | **include** regular-expression

## 14-70 show ip bgp flap-statistics prefix mask longer-prefixes

## 14-71 show ip bgp flap-statistics prefix mask longer-prefixes | begin regular-expression

**14-72 show ip bgp flap-statistics** prefix mask **longer-prefixes** |  
**exclude** regular-expression

**14-73 show ip bgp flap-statistics** prefix mask **longer-prefixes** |  
**include** regular-expression

## 14-74 **show ip bgp flap-statistics filter-list** list-number



**14-75 show ip bgp flap-statistics filter-list** *list-number* | **begin**  
*regular-expression*

**14-76 show ip bgp flap-statistics filter-list** *list-number* | **exclude**  
*regular-expression*

**14-77 show ip bgp flap-statistics filter-list** *list-number* | **include**  
*regular-expression*

## 14-78 show ip bgp flap-statistics quote-regexp quoted-line

**14-79 show ip bgp flap-statistics quote-regexp** quoted-line | **begin**  
regular-expression

**14-80 show ip bgp flap-statistics quote-regexp** quoted-line | **exclude**  
regular-expression

**14-81 show ip bgp flap-statistics quote-regexp** `quoted-line` | **include**  
regular-expression

## 14-82 show ip bgp flap-statistics regexp regular-expression



## 14-83 show ip bgp flap-statistics | begin regular-expression

## 14-84 show ip bgp flap-statistics | exclude regular-expression



## 14-85 show ip bgp flap-statistics | include regular-expression

Syntax Description:

- 
- prefix— IP address of the prefix to display.
- 
- 
- mask— Display prefix with a specific network mask.
- 
- 
- mask-length— Bit length of the mask.
- 
- 
- line— Regular expression. See [Appendix B](#).
- 
- 
- list-number— Regular expression access list number. The range of values is 1 to 199.
- 
- 
- quoted-line— Regular expression in quotes.
- 

Purpose: To display flap statistics for routes in the BGP table.

Cisco IOS Software Release: 11.0

### Example: Display BGP Flap Statistics

A route flap occurs when a route is withdrawn and then later readvertised. This usually happens when an interface is going from the UP state to the DOWN state repeatedly. You can view the number of times a route has flapped using the show ip bgp flap-statistics command. This assumes that BGPP dampening has been enabled on the router.

```
rtrA#show ip bgp flap-statistics
```

```
BGP table version is 9, local router ID is 10.1.1.2
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```



## 14-86 show ip bgp inconsistent-as

## 14-87 show ip bgp inconsistent-as | begin line

## 14-88 show ip bgp inconsistent-as | exclude line



## 14-89 show ip bgp inconsistent-as | include line

Syntax Description:

- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display routes with inconsistent originating autonomous systems.

Cisco IOS Software Release: 11.0

### Example: Display BGP Routes Having an Inconsistent Originated AS

Routes that have the same next-hop address should come from the same AS. When this is not the case, inconsistent AS information exists. The inconsistent AS information can be displayed using the show ip bgp inconsistent-as command.

```
rtrA#show ip bgp inconsistent-as
```

```
BGP table version is 3, local router ID is 172.17.1.1
```

```
Status codes: s suppressed, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 199.1.1.0	172.17.1.2	0			0 1 2 ?
*>	172.17.1.2	0			0 3 ?

## 14-90 show ip bgp neighbors

## 14-91 show ip bgp neighbors | begin line

## 14-92 show ip bgp neighbors | exclude line

## 14-93 show ip bgp neighbors | include line

## 14-94 show ip bgp neighbors ip-address advertised-routes

## 14-95 show ip bgp neighbors ip-address dampened-routes

## 14-96 show ip bgp neighbors ip-address flap-statistics



## 14-97 show ip bgp neighbors ip-address paths

## 14-98 show ip bgp neighbors ip-address paths line

## 14-99 show ip bgp neighbors ip-address received-routes



## 14-100 show ip bgp neighbors ip-address routes

Syntax Description:

- 
- ip-address— IP address of the BGP neighbor.
- 
- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display routes with inconsistent originating autonomous systems. Soft inbound reconfiguration needs to be enabled before you use the received-routes form of this command.

Cisco IOS Software Release: 10.0. The received-routes keyword was added in Cisco IOS Software Release 11.2.

### Example: Display Information for a Specific BGP Neighbor

The state of a BGP neighbor can be examined by using the show ip bgp neighbors command. This is usually the first command to use when debugging a BGP connection.

```
rtrA#show ip bgp neighbors 172.17.1.2
```

```
BGP neighbor is 172.17.1.2, remote AS 1, external link
```

```
  BGP version 4, remote router ID 199.172.15.1
```

```
  BGP state = Established, up for 1w1d
```

```
  Last read 00:00:43, hold time is 180, keepalive interval is 60 seconds
```

```
  Neighbor capabilities:
```

```
    Route refresh: advertised
```

```
    Address family IPv4 Unicast: advertised and received
```

```
  Received 13174 messages, 0 notifications, 0 in queue
```

```
  Sent 13168 messages, 0 notifications, 0 in queue
```

```
  Route refresh request: received 0, sent 0
```

```
  Minimum time between advertisement runs is 30 seconds
```



## 14-101 show ip bgp paths

## 14-102 show ip bgp paths line



## 14-103 show ip bgp paths line

## 14-104 show ip bgp paths line

## 14-105 show ip bgp paths | begin line

## 14-106 show ip bgp paths | exclude line



## 14-107 show ip bgp paths | include line

Syntax Description:

- 
- ip-address— IP address of the BGP neighbor.
- 
- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display all the BGP paths in the database.

Cisco IOS Software Release: 10.0

### Example: Display BGP Path Information

The AS paths contained in the BGP database can be displayed using the show ip bgp paths command.

```
rtrA#show ip bgp paths
```

Address	Hash	Refcount	Metric	Path
0x125E94	0	4	0	i
0x996F4	2	1	0	1 i
0x125D6C	2	2	0	1 i

The following list explains some of the fields that appear in the preceding output from the show ip bgp paths command.

- 
- Address— Memory address of where the path is stored.
- 
- 
- Hash— Hash bucket where the path is stored.
- 
- 
- Refcount— Number of prefixes using that path.



## 14-108 show ip bgp peer-group



## 14-109 **show ip bgp peer-group** peer-group-name

## 14-110 **show ip bgp peer-group** peer-group-name **summary**

Syntax Description:

- 
- peer-group-name— Name of the peer group.
- 

Purpose: To display information regarding BGP peer groups.

Cisco IOS Software Release: 11.0

### **Example: Display BGP Peer Group Information**

To view the parameters of a BGP peer group, use the `show ip bgp peer-group` command.

```
rtrA#show ip bgp peer-group
```

```
BGP neighbor is demo, peer-group leader
```

```
  BGP version 4
```

```
  Minimum time between advertisement runs is 5 seconds
```

```
  Incoming update AS path filter list is 30
```

```
  Outgoing update AS path filter list is 40
```

```
  Route map for outgoing advertisements is adjust-weight
```

## 14-111 show ip bgp summary

## 14-112 show ip bgp summary | begin line

## 14-113 show ip bgp summary | exclude line



## 14-114 show ip bgp summary | include line

Syntax Description:

- 
- line— Regular expression. See [Appendix B](#).
- 

Purpose: To display the status of BGP connections.

Cisco IOS Software Release: 10.0. The PfxRcd and Admin fields were added in version 12.0.

### Example: Display a Summary for All BGP Connections

Display the entire BGP table:

```
rtrA#show ip bgp summary
```

```
BGP router identifier 200.1.4.1, local AS number 2
```

```
BGP table version is 14, main routing table version 14
```

```
7 network entries and 7 paths using 931 bytes of memory
```

```
3 BGP path attribute entries using 156 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 activity 193/657 prefixes, 193/186 paths, scan interval 15 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.1.1.2	4	2	7584	7590	14	0	0	5d06h	0
172.17.1.2	4	1	13359	13353	14	0	0	1w1d	3

The following list explains some of the fields that appear in the preceding output from the show ip bgp summary command.







# Chapter 15. BGP clear Commands

- [Section 15-1. clear ip bgp \\*](#)
- 
- [Section 15-2. clear ip bgp \\* soft](#)
- 
- [Section 15-3. clear ip bgp \\* soft in](#)
- 
- [Section 15-4. clear ip bgp \\* soft out](#)
- 
- [Section 15-5. clear ip bgp \\* soft in out](#)
- 
- [Section 15-6. clear ip bgp AS-number](#)
- 
- [Section 15-7. clear ip bgp AS-number soft](#)
- 
- [Section 15-8. clear ip bgp AS-number soft in](#)
- 
- [Section 15-9. clear ip bgp AS-number soft out](#)
- 
- [Section 15-10. clear ip bgp AS-number soft in out](#)
- 
- [Section 15-11. clear ip bgp neighbor-ip-address](#)
- 
- [Section 15-12. clear ip bgp neighbor-ip-address soft](#)
- 
- [Section 15-13. clear ip bgp neighbor-ip-address soft in](#)
- 
- [Section 15-14. clear ip bgp neighbor-ip-address soft out](#)
- 
- [Section 15-15. clear ip bgp neighbor-ip-address soft in out](#)
- 
- [Section 15-16. clear ip bgp peer-group peer-group-name](#)



## 15-1 clear ip bgp \*

## 15-2 clear ip bgp \* soft

## 15-3 clear ip bgp \* soft in

## 15-4 clear ip bgp \* soft out

## 15-5 clear ip bgp \* soft in out



## 15-6 clear ip bgp AS-number

## 15-7 clear ip bgp AS-number soft

## 15-8 clear ip bgp AS-number soft in

## 15-9 clear ip bgp AS-number soft out

## 15-10 clear ip bgp AS-number soft in out

## 15-11 clear ip bgp neighbor-ip-address

## 15-12 clear ip bgp neighbor-ip-address soft

## 15-13 clear ip bgp neighbor-ip-address soft in



## 15-14 clear ip bgp neighbor-ip-address soft out

## **15-15 clear ip bgp neighbor-ip-address soft in out**

## **15-16 clear ip bgp peer-group** peer-group-name

**15-17 clear ip bgp peer-group** peer-group-name **soft**

## **15-18 clear ip bgp peer-group** peer-group-name **soft in**

**15-19 clear ip bgp peer-group** peer-group-name **soft out**



## 15-20 clear ip bgp peer-group peer-group-name soft in out

Syntax Description:

- 
- AS-number— Resets all neighbors in the specified autonomous system.
- 
- 
- neighbor-ip-address— Resets the specified BGP neighbor.
- 
- 
- peer-group-name— Resets all neighbors in the specified peer group.
- 

Purpose: To reset a BGP connection. BGP neighbor connections can be reset based on the neighbor's IP address, the neighbor's autonomous system number, or the peer group name. You must reset a BGP connection when any of the following have been modified or added to:

- 
- BGP access list
- 
- 
- BGP distribute list
- 
- 
- BGP route map
- 
- 
- BGP timers
- 
- 
- BGP weights
- 
- 
- BGP administrative distance
- 

Clearing a BGP connection without the soft keyword causes the neighbor relationship to transition from Established to Idle. The neighbor relationship is then reestablished, and the new routing policies take effect. If the soft keyword is used, the session is not reset, and all routing updates are resent. When you use the soft





## 15-21 clear ip bgp dampening

## 15-22 clear ip bgp dampening prefix mask

Syntax Description:

- 
- prefix— Prefix of the dampened route to clear.
- 
- 
- mask— Mask of the dampened route to clear.
- 

Purpose: To clear dampening information for all dampened routes or for a specific dampened route. Suppressed routes will be unsuppressed.

Cisco IOS Software Release: 11.0

## 15-23 clear ip bgp flap-statistics

## 15-24 clear ip bgp flap-statistics prefix mask

## **15-25 clear ip bgp flap-statistics filter-list** list-number

## 15-26 clear ip bgp flap-statistics regexp regular-expression

## 15-27 clear ip bgp neighbor-ip-address flap-statistics

Syntax Description:

- 
- prefix— Prefix of the dampened route to clear.
- 
- 
- mask— Mask of the dampened route to clear.
- 
- 
- list-number— Number of the IP access list used to identify prefixes to be cleared.
- 
- 
- regular-expression— Clear statistics for routes matching the regular expression.
- 
- 
- neighbor-ip-address— Clear flap statistics of routes received from the neighbor.
- 

Purpose: To clear the flap statistics for all routes or for specific routes. See [section 3-9](#) through [3-12](#) regarding the configuration of BGP dampening.

Cisco IOS Software Release: 11.0





# Chapter 16. BGP debug Commands

- [Section 16-1. debug ip bgp](#)
- 
- [Section 16-2. debug ip bgp neighbor-ip-address updates](#)
- 
- [Section 16-3. debug ip bgp neighbor-ip-address updates access-list-number](#)
- 
- [Section 16-4. debug ip bgp dampening](#)
- 
- [Section 16-5. debug ip bgp dampening access-list-number](#)
- 
- [Section 16-6. debug ip bgp events](#)
- 
- [Section 16-7. debug ip bgp keepalives](#)
- 
- [Section 16-8. debug ip bgp in neighbor-ip-address updates](#)
- 
- [Section 16-9. debug ip bgp in neighbor-ip-address updates access-list-number](#)
- 
- [Section 16-10. debug ip bgp out neighbor-ip-address updates](#)
- 
- [Section 16-11. debug ip bgp out neighbor-ip-address updates access-list-number](#)
- 
- [Section 16-12. debug ip bgp updates](#)
- 
- [Section 16-13. debug ip bgp updates in](#)
- 
- [Section 16-14. debug ip bgp updates out](#)
- 
- [Section 16-15. debug ip bgp updates access-list-number](#)
- 
- [Section 16-16. debug ip bgp updates access-list-number in](#)





## 16-1 debug ip bgp

Syntax Description: This command has no arguments.

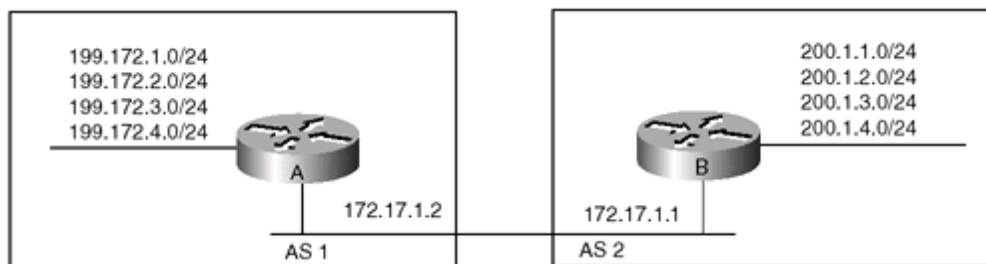
Purpose: Use to debug the formation of the BGP neighbor relationship.

Cisco IOS Software Release: 10.0

### Configuration Example

In [Figure 16-1](#), Router B has been configured with the wrong AS number for Router A.

**Figure 16-1. Use debug ip bgp to Determine Configuration Errors**



Router A

```
router bgp 1
```

```
neighbor 172.17.1.1 remote-as 2
```

---

Router B

```
router bgp 2
```

```
neighbor 172.17.1.2 remote-as 1
```

Use debug ip bgp to identify the problem with the formation of the neighbor relationship between Routers A and B:

```
rttrB#debug ip bgp
```

```
rttrB#clear ip bgp *
```



## 16-2 debug ip bgp neighbor-ip-address updates





## 16-3 debug ip bgp neighbor-ip-address updates access-list-number

Syntax Description:

- 
- neighbor-ip-address— BGP neighbor's IP address.
- 
- 
- access-list-number— IP access list number.
- 

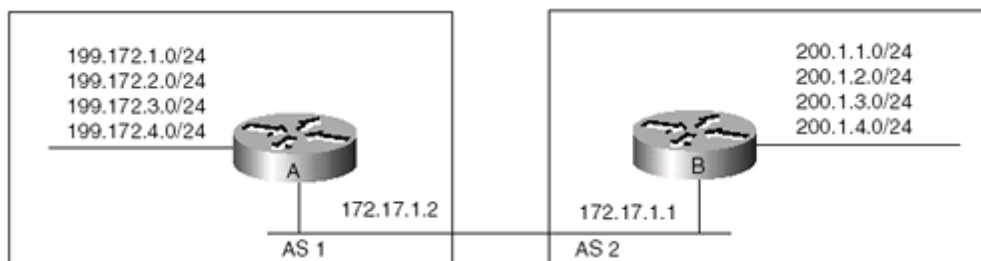
Purpose: To debug BGP updates from a particular neighbor. The first form debugs all updates from the neighbor. The second form can be used to debug specific updates from the neighbor. The IP access list number can be in the range 1 to 199 or 1300 to 2699.

Cisco IOS Software Release: 10.0

### Example 1: Debug All Updates to and from a Particular Neighbor

In [Figure 16-2](#), Router A is advertising four prefixes to Router B, and Router B is advertising four prefixes to Router A. Debug all the updates received from and sent to Router A.

**Figure 16-2. Debug All Updates from a Neighbor**



Router A

```
interface loopback 1
```

```
ip address 199.172.1.1 255.255.255.0
```

```
!
```

```
interface loopback 2
```

```
ip address 199.172.2.1 255.255.255.0
```



## 16-4 debug ip bgp dampening



## 16-5 debug ip bgp dampening access-list-number

Syntax Description:

- 
- access-list-number— IP access list number.
- 

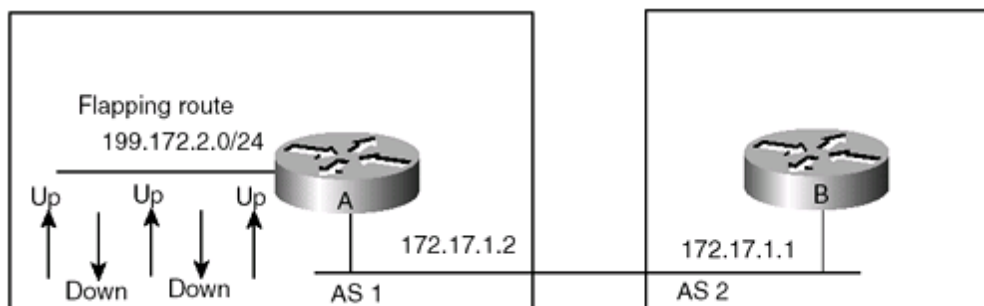
Purpose: To debug BGP events associated with dampening. The first form debugs all dampening events. The second form debugs only dampening events associated with routes identified by the IP access list. The IP access list number can be in the range from 1 to 199 or 1300 to 2699. Although it isn't necessary, use logging to store the results so that you can review them later.

Cisco IOS Software Release: 10.0

### Example: Debug All BGP Dampening Events

In [Figure 16-3](#), Router A is advertising network 199.172.2.0/24 via BGP to Router B. Dampening is enabled on Router B, and the 199.172.2.0/24 network on Router A is flapped by alternately bringing the interface up and down.

Figure 16-3. Scenario for Debugging BGP Dampening



Router A

```
interface loopback 1
```

```
ip address 199.172.2.0 255.255.255.0
```

```
router bgp 1
```

```
neighbor 172.17.1.1 remote-as 2
```





## 16-6 debug ip bgp events

Syntax Description: This command has no arguments.

Purpose: To debug events relating to the BGP neighbor relationship.

Cisco IOS Software Release: 10.0

### Example: Debug the Formation of the Neighbor Relationship

Enable BGP event debugging, and then clear the connection between the BGP neighbors:

```
rtrA#debug ip bgp events
BGP events debugging is on
rtrA#clear ip bgp *
rtrA#
1w3d: BGP: reset all neighbors due to User reset
1w3d: BGP: 172.17.1.1 went from Active to Idle
1w3d: BGP: 172.17.1.1 went from Idle to Active
1w3d: BGP: scanning routing tables
1w3d: BGP: 172.17.1.1 went from Active to Idle
1w3d: BGP: 172.17.1.1 went from Idle to Connect
1w3d: BGP: 172.17.1.1 went from Connect to OpenSent
1w3d: BGP: 172.17.1.1 went from OpenSent to OpenConfirm
1w3d: BGP: 172.17.1.1 went from OpenConfirm to Established
1w3d: BGP: 172.17.1.1 computing updates, neighbor version 0, table version 5,
starting at 0.0.0.0
1w3d: BGP: 172.17.1.1 update run completed, ran for 0ms, neighbor version 0,
start version 5, throttled to 5, check point net 0.0.0.0
```





## 16-7 debug ip bgp keepalives

Syntax Description: This command has no arguments.

Purpose: To debug keepalive messages sent between BGP neighbors. To configure the keepalive interval, see [sections 8-31](#) and [13-1](#).

Cisco IOS Software Release: 10.0

### Example: Debug BGP Keepalive Messages

Enable BGP keepalive message debugging between any two BGP neighbors:

```
rtrB#debug ip bgp keepalives
```

```
BGP keepalives debugging is on
```

```
rtrB#
```

```
1d06h: BGP: 172.17.1.2 sending KEEPALIVE
```

```
1d06h: BGP: 172.17.1.2 KEEPALIVE rcvd
```

## 16-8 debug ip bgp in neighbor-ip-address updates

## **16-9 debug ip bgp in** neighbor-ip-address **updates** access-list-number

## 16-10 debug ip bgp out neighbor-ip-address updates

## 16-11 debug ip bgp out neighbor-ip-address **updates** access-list-number

## 16-12 debug ip bgp updates

## 16-13 debug ip bgp updates in



## 16-14 debug ip bgp updates out

# 16-15 debug ip bgp updates `access-list-number`

## 16-16 debug ip bgp updates access-list-number in



## 16-17 debug ip bgp updates access-list-number out

Syntax Description:

- 
- neighbor-ip-address— BGP neighbor's IP address.
- 
- 
- access-list-number— IP access list number.
- 

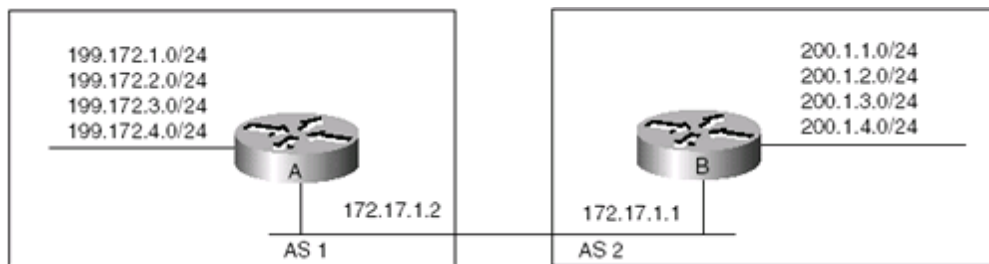
Purpose: To debug BGP updates. The IP access list number can be in the range 1 to 199 or 1300 to 2699.

Cisco IOS Software Release: 10.0

### Example 1: Debug All BGP Updates

In [Figure 16-4](#), Router A is advertising four prefixes to Router B, and Router B is advertising four prefixes to Router A. Debug all the updates on Router B.

Figure 16-4. Debug All Updates from a Neighbor



Router A

```
interface loopback 1
```

```
ip address 199.172.1.1 255.255.255.0
```

```
!
```

```
interface loopback 2
```

```
ip address 199.172.2.1 255.255.255.0
```

```
!
```

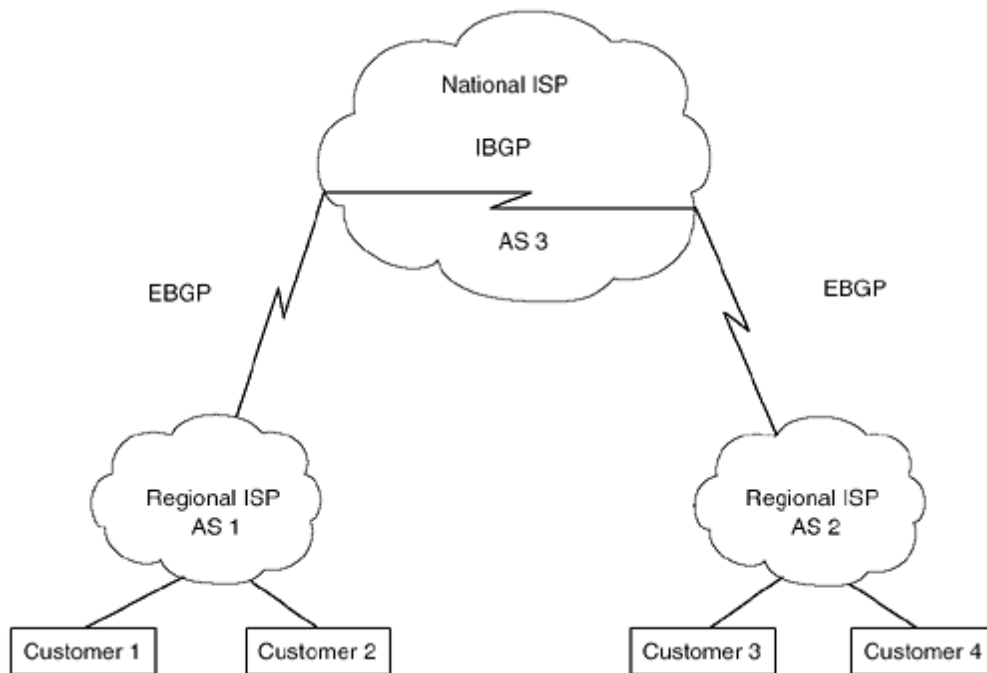




# appendix A. RFC 1771: Border Gateway Protocol 4

The objective of this appendix is to present the concepts and terminology contained in RFC 1771 in order to help you understand the operation of BGP. Many of the details have been omitted, such as packet format and contents. These topics are of interest to developers of the code that runs on a BGP-capable router. Border Gateway Protocol (BGP) is an interauto-nomous system routing protocol. An autonomous system (AS) is a network or group of networks that are under a common administration and that have common routing policies. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISPs). Customer networks, such as universities and corporations, usually employ an Interior Gateway Protocol (IGP) such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) for the exchange of routing information within their networks. Customers connect to ISPs, and ISPs use BGP to exchange customer and ISP routes. When BGP is used between autonomous systems, the protocol is called External BGP (EBGP). If a service provider uses BGP to exchange routes within its AS, the protocol is called Interior BGP (IBGP). [Figure A-1](#) illustrates this distinction.

**Figure A-1. External and Interior BGP**



BGP is a very robust and scalable routing protocol, as evidenced by the fact that BGP is the routing protocol employed on the Internet. Currently, the Internet BGP routing tables have more than 90,000 routes. In order to achieve scalability at this level, BGP uses many route parameters, called attributes, to define routing policies and maintain a stable routing environment.

In addition to BGP attributes, BGP uses Classless Interdomain Routing (CIDR) to reduce the size of the







# BGP Attributes

Routes learned via BGP have associated properties that are used to determine the best route to a destination when multiple paths exist. These properties are called BGP attributes. An understanding of how BGP attributes influence route selection is required for the design of robust networks. This section describes the attributes that BGP uses in the route selection process:

- 
- Weight
  - 
  -
- Local preference
  - 
  -
- Multi-exit discriminator
  - 
  -
- Origin
  - 
  -
- AS\_path
  - 
  -
- Next-hop
  - 
  -
- Community
  -

## Weight Attribute

Weight is a Cisco-defined attribute that is local to a router. The weight attribute is not advertised to neighboring routers. If the router learns about more than one route to the same destination, the route with the highest weight is preferred. In [Figure A-2](#), Router A is receiving an advertisement for network 172.16.1.0 from Routers B and C. When Router A receives the advertisement from Router B, the associated weight is set to 50. When Router A receives the advertisement from Router C, the associated weight is set to 100. Both paths for network 172.16.1.0 will be in the BGP routing table with their respective weights. The route with the higher weight will be installed in the IP routing table.

Figure A-2 BGP Weight Attribute





# BGP Path Selection

BGP can receive multiple advertisements for the same route from multiple sources. BGP selects only one path as the best path. After the path is selected, BGP puts it in the IP routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order listed, to select a path for a destination:

- 
- Ignore a route if the next hop is not known.
- 
- 
- Ignore IBGP routes that are not synchronized.
- 
- 
- Prefer the route with the largest weight.
- 
- 
- Prefer the route with the largest local preference.
- 
- 
- Prefer the route that was locally originated.
- 
- 
- Prefer the route with the shortest AS path. If you're using `bgp bestpath as-path ignore`, skip this step. When you use the `as-set` option for aggregated routes, it counts as 1 regardless of the number of AS entries in the set. Confederation sub-AS numbers are not used to determine the AS-path length.
- 
- 
- Prefer the route with the lowest origin (IGP < EGP < Incomplete).
- 
- 
- Prefer the route with the lowest MED. This comparison is only between routes advertised by the same external AS.
- If you're using `bgp always-compare-med`, compare MEDs for all paths. If used, this command needs to be configured on every BGP router in the AS.
- 
- If you're using `bgp bestpath med-confed`, the MEDs are compared only for routes that have an AS confederation sequence in their AS-path attribute.
-



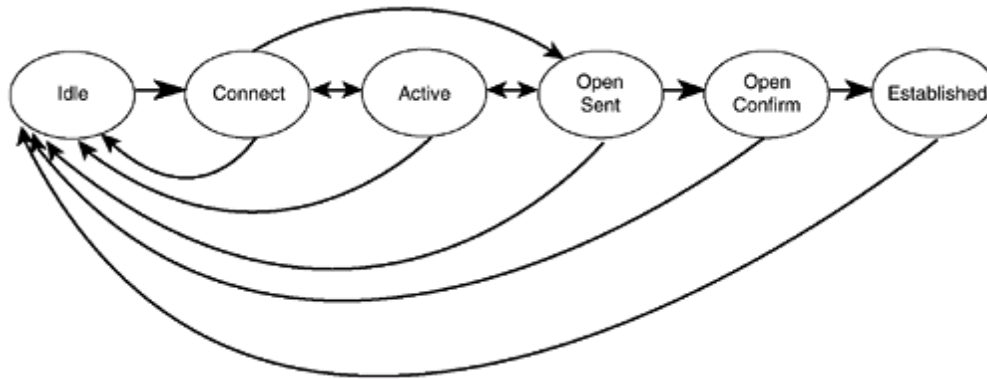




# Forming a BGP Connection

BGP uses TCP/IP as its underlying transport and uses TCP port 179. BGP speakers transition through various states during the establishment of a neighbor relationship, as shown in [Figure A-10](#).

**Figure A-10. States in the Formation of a BGP Neighbor Relationship**



The Idle state usually represents the condition in which BGP has not been enabled on the router, or a particular neighbor relationship has been shut down. When a connection is initiated through either router configuration or the removal of the shut-down state for a particular neighbor, BGP moves to the Connect state.

In the Connect state, BGP waits for a TCP connection to be established with the remote neighbor. If the TCP connection is established, an Open message is sent, and the Open Sent state is entered. If the TCP connection is not established, the Active state is entered. While in the Active state, the router continues to listen for a connection.

In the Open Sent state, the router waits for an Open message from the remote neighbor. If there are errors in the Open message, a notification message is sent to the neighbor, and the local router enters the Idle state.

If the Open message contains no errors, a keepalive message is sent, and the holdtime is set to the negotiated value. The Open Confirm state is then entered. If the BGP neighbors are in the same AS, the BGP connection is internal. If the BGP neighbors are in different autonomous systems, the connection is external.

In the Open Confirm state, BGP waits for a keepalive or notification message. If a keepalive message is received, the connection moves to the Established state. If a keepalive message is not received before the hold timer expires, a notification message is sent, and the state transitions to Idle.

In the Established state, the BGP neighbors can exchange update, notification, and keepalive messages. Whenever BGP changes its state from Established to Idle, it closes the BGP and TCP connection, releases





# Appendix B. Regular Expressions

A regular expression is a sequence of one or more characters that can be used for two general purposes. The first purpose is to reduce the output from various show commands. For example, if you wanted to view the routing protocols that are active on a router, you could use show running-config and scroll through the output until you came to the section of the configuration that contained the routing configurations. Or, you could use an output modifier with a regular expression to display only those lines containing a router configuration command:

```
rtrA#show running-config | include router

router eigrp 1

router bgp 2
```

This code uses the output modifier include and the regular expression router to reduce the amount of output from the show running-config command. Each line of output from this command is examined, and if the pattern router is contained in the line, the line is displayed. You could use the same regular expression with a different output modifier to display the configuration, starting with the router configurations and everything that follows:

```
p2#show running-config | begin router

router eigrp 1

  network 10.0.0.0

  network 172.17.0.0

!

router bgp 2

  network 200.1.1.0

  network 200.1.2.0

  neighbor 10.1.1.2 remote-as 2

  neighbor 172.17.1.2 remote-as 1

!

line con 0

  exec-timeout 0 0

  transport input none
```





# appendix C. Route Map Logic

A route map is an extremely powerful and versatile tool for route filtering and attribute manipulation. In regards to BGP, route maps are used in the following commands:

```
aggregate-address address mask advertise-map route-map-name
```

```
aggregate-address address mask as-set route-map-name
```

```
aggregate-address address mask attribute-map route-map-name
```

```
aggregate-address address mask route-map route-map-name
```

```
aggregate-address address mask suppress-map route-map-name
```

```
bgp dampening route-map route-map-name
```

```
neighbor ip-address advertise-map route-map-name non-exist-map route-map-name
```

```
neighbor ip-address default-originate route-map route-map-name
```

```
neighbor ip-address route-map route-map-name in
```

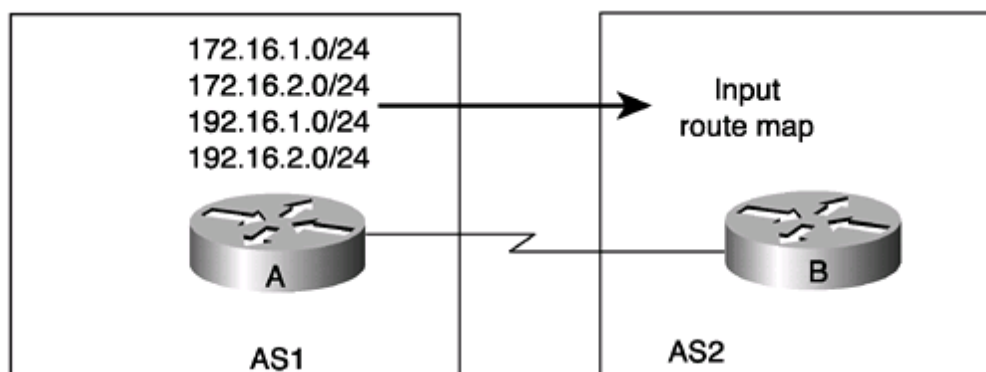
```
neighbor ip-address route-map route-map-name out
```

```
neighbor ip-address unsuppress-map route-map-name
```

```
redistribute protocol route-map route-map-name
```

These commands allow you to filter routes, manipulate BGP attributes, or both. The logic of route maps is demonstrated in [Figure C-1](#).

**Figure C-1. Scenario for Illustrating Route Map Logic**









## Form 1: permit/permit

With form 1, the recommended form, if a match is made, accept the route, set the attribute (if you're using a set clause), and exit the route map. Otherwise, continue to the next route-map statement:

```
access-list 1 permit ip-address/mask
```

or

```
ip as-path access-list number permit regular-expression
```

or

```
ip prefix-list name permit prefix/length
```

or

```
ip community-list number permit community-number
```

```
route-map name permit sequence-number
```

```
  match ip address access-list-number
```

or

```
  match as-path access-list-number
```

or

```
  match ip address prefix-list list-name
```





## Form 2: permit/deny

If a match is made, deny the route and exit the route map.

```
access-list 1 permit ip-address/mask
```

or

```
ip as-path access-list number permit regular-expression
```

or

```
ip prefix-list name permit prefix/length
```

or

```
ip community-list number permit community-number
```

```
route-map name deny sequence-number
```

```
  match ip address access-list-number
```

or

```
  match as-path access-list-number
```

or

```
  match ip address prefix-list list-name
```

or





## Form 3: deny/permit

If a match is made, deny the route and exit the route map:

```
access-list 1 deny ip-address/mask
```

or

```
ip as-path access-list number deny regular-expression
```

or

```
ip prefix-list name deny prefix/length
```

or

```
ip community-list number deny community-number
```

```
route-map name permit sequence-number
```

```
  match ip address access-list-number
```

or

```
  match as-path access-list-number
```

or

```
  match ip address prefix-list list-name
```

or







## Form 4: deny/deny

If a match is made, accept the route and continue to the next route-map statement:

```
access-list 1 deny ip-address/mask
```

or

```
ip as-path access-list number deny regular-expression
```

or

```
ip prefix-list name deny prefix/length
```

or

```
ip community-list number deny community-number
```

```
route-map name permit sequence-number
```

```
  match ip address access-list-number
```

or

```
  match as-path access-list-number
```

or

```
  match ip address prefix-list list-name
```

or



← PREVIOUS

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

← PREVIOUS

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

prefix-list out option:neighbor command

[options:neighbor command:prefix-list out](#) [2nd](#) [3rd](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[accepted prefixes field \(show ip bgp neighbors command\)](#)

access lists:standard:configuring

configuring:access lists:standard

[standard access lists:configuring](#) [2nd](#)

[ACK hold field \(show ip bgp neighbors command\)](#)

adding:descriptions to BGP configurations

inserting:descriptions in BGP configurations

[description option:neighbor command](#) [2nd](#)

[Address family IPv4 Multicast field \(show ip bgp neighbors command\)](#)

[Address family IPv4 Unicast field \(show ip bgp neighbors command\)](#)

administrative distance:best path, selecting

best path:selecting

[selecting:best path](#) [2nd](#) [3rd](#)

[advertisement runs field \(show ip bgp neighbors command\)](#)

[advertising routes](#)

advertising:directly connected networks

[directly connected networks:advertising](#) [2nd](#) [3rd](#)

aggregate-address command

[commands:aggregate-address](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#) [12th](#) [13th](#) [14th](#) [15th](#)  
[16th](#) [17th](#) [18th](#) [19th](#) [20th](#) [21st](#) [22nd](#)

[BGP table:adding aggregate entries;adding:aggregate entries to BGP table;inserting:aggregate entries](#)

[2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

aggregate-address command:as-set advertise-map option

commands:aggregate-address:as-set advertise-map option

[as-set advertise-map option:aggregate-address command;options:aggregate-address command:as-set](#)

[adver](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

aggregate-address command:as-set option

commands:aggregate-address:as-set option

[as-set option:aggregate-address command;options:aggregate-address command:as-set](#) [2nd](#) [3rd](#) [4th](#)

[5th](#) [6th](#)

aggregate-address command:attribute-map option

commands:aggregate-address:attribute-map option

[attribute-map option:aggregate-address command;options:aggregate-address command:attribute-map](#)

aggregate-address command:route-map option

commands:aggregate-address:route-map option

[route-map option:aggregate-address command;options:aggregate-address command:route-map](#) [2nd](#)

[3rd](#) [4th](#) [5th](#) [6th](#)

aggregate-address command:summary-only option

commands:aggregate-address:summary-only option

[summary-only option:aggregate-address command;options:aggregate-address command:summary-only](#)

[2nd](#) [3rd](#) [4th](#)

aggregate-address command:suppress-map option

commands:aggregate-address:suppress-map option

[suppress-map option:aggregate-address command;options:aggregate-address command:suppress-map](#)

[2nd](#) [3rd](#) [4th](#)

aggregation:attributes:modifying

modifying:aggregate route attributes

[attributes:of aggregate routes:modifying](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

aggregation:based on subset of prefixes from different ASs

[prefixes:subset from different ASs:aggregation](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)







backdoor option:network command

network command:backdoor option

[commands:network:backdoor option](#) [2nd](#) [3rd](#)

[best path](#)

[best path selection](#)

best path:selecting

[selecting:best path](#) [2nd](#)

bgp always-compare-med command

[commands:bgp always-compare-med](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

bgp bestpath as-path ignore command

[commands:bgp bestpath as-path ignore](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

bgp bestpath med confed command

[commands:bgp bestpath med confed](#) [2nd](#) [3rd](#)

bgp bestpath med missing-as-worst command

[commands:bgp bestpath med missing-as-worst](#) [2nd](#) [3rd](#)

bgp client-to-client reflection command

[commands:bgp client-to-client reflection](#) [2nd](#)

bgp cluster-id command

[commands:bgp cluster-id](#)

bgp confederation peers command

[commands:bgp confederation peers](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

bgp dampening route-map command

[commands:bgp dampening route-map](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

bgp default local-preference command

[commands:bgp default local-preference](#) [2nd](#)

bgp deterministic-med command

[commands:bgp deterministic-med](#)

bgp fast-external-failover command

[commands:bgp fast-external-failover](#) [2nd](#) [3rd](#) [4th](#)

bgp log-neighbor-changes command

[commands:bgp log-neighbor-changes](#)

[BGP neighbor field \(show ip bgp neighbors command\)](#)

[BGP router identifier field \(show ip bgp summary command\)](#)

bgp router-id command

[commands:bgp router-id](#) [2nd](#)

[BGP state field \(show ip bgp neighbors command\)](#)

[BGP table version field \(show ip bgp flap-statistics command\)](#)

[BGP table version field \(show ip bgp neighbors command\)](#)

[BGP table version field \(show ip bgp output\)](#)

[BGP table version field \(show ip bgp summary command\)](#)

[BGP version field \(show ip bgp neighbors command\)](#)

BGP:version, locking down

[locking down BGP version](#) [2nd](#)





[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

## [CIDR \(Classless Interdomain Routing\)](#)

classful addresses

[addresses:classful](#)

[IP addresses:classful](#)

clear commands

[BGP clear commands](#)

[commands:BGP clear](#) [2nd](#) [3rd](#) [4th](#)

client-to-client reflection:configuring

[route reflectors:configuring](#)

[configuring:route reflection](#) [2nd](#)

cluster IDs

[route reflectors:clusters](#)

[speakers:route reflectors:clusters](#)

community attribute

[attributes:community](#) [2nd](#) [3rd](#)

[Community attribute field \(show ip bgp neighbors command\)](#)

community attribute:modifying

[modifying:community attribute](#) [2nd](#) [3rd](#)

community attributes:NO-EXPORT community value, configuring

[NO-EXPORT community value:configuring](#)

[attribute:community:NO-EXPORT community value, configuring](#) [2nd](#) [3rd](#)

conditional default routes:configuring

[configuring:default routes:conditional](#)

[default routes:conditional:configuring;advertising:conditional default routes](#) [2nd](#) [3rd](#)

configuring:updates:minimum interval

[updates:minimum interval:configuring](#)

[minimum interval between updates:configuring](#) [2nd](#)

## [Connect state](#)

connected routes:redistribution

[static routes:redistribution](#)

[EIGRP:learned routes:redistribution;learned routes:EIGRP:redistribution](#) [2nd](#) [3rd](#) [4th](#)

connected routes:redistribution:verifying

[verifying:redistribution of connected routes](#)

[redistribution:connected routes:verifying](#) [2nd](#)

[Connection state field \(show ip bgp neighbors command\)](#)

[Connections established field \(show ip bgp neighbors command\)](#)

connections:BGP:resetting

[resetting:BGP connections](#)

connections:clearing

[clearing:connections](#)

[resetting:connections](#)

connections:forming

[forming:connections](#)

[creating:connections](#) [2nd](#)

connections:status, displaying

[displaying:BGP connection status](#)

[viewing:BGP connection status](#)

customizing:dampening parameters

[parameters:route dampening:customizing](#)

[route dampening:parameters:customizing;flapping routes:dampening:customized parameters](#) [2nd](#)







[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

dampened routes

[clearing information](#)

dampening events

[debugging](#) [2nd](#)

[Datagrams Rcvd field \(show ip bgp neighbors command\)](#)

debug ip bgp command

[commands:debug ip bgp](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#)

default local preference:configuring

[configuring:local preference:default value](#)

[local preference:default value:configuring](#) [2nd](#) [3rd](#)

default routes:advertising

[advertising:default routes](#) [2nd](#) [3rd](#) [4th](#)

default routes:configuring

[configuring:default routes](#) [2nd](#) [3rd](#)

default-information originate command

[commands:default-information originate](#) [2nd](#) [3rd](#)

default-metric command

[commands:default-metric](#) [2nd](#) [3rd](#)

[delrecvwnd field \(show ip bgp neighbors command\)](#)

devices:icons for

[icons:network devices](#)

displaying

[specified prefix in BGP routing table](#) [2nd](#)

displaying:AS path access lists:corresponding dampened prefixes

[AS path access lists:corresponding dampened prefixes, displaying](#)

[access lists:corresponding dampened prefixes, displaying](#)

displaying:BGP routing table

[viewing:BGP routing table](#)

[routing table:displaying;entries:BGP routing table:displaying](#) [2nd](#)

displaying:natural masks in routing table

[viewing:natural masks in routing table](#)

[routing table:natural masks:displaying;natural masks:displaying in routing table](#)

displaying:neighbor status change events on console

[viewing:neighbor status change events on console](#)

displaying:prefixes belonging to specified community

[communities:displaying corresponding prefixes](#)

[prefixes:corresponding communities:displaying;viewing:prefixes and corresponding communities;routing](#)

[2nd](#) [3rd](#)

displaying:routing table entries:flap statistics

[routing table:flap statistics, displaying](#)

[flap statistics:displaying;viewing:routing table entries:flap statistics](#) [2nd](#)

distance command

[commands:distance](#)

[administrative distance:modifying;modifying:administrative distance](#) [2nd](#) [3rd](#) [4th](#)

distribute-list command:

[commands:distribute-list:See neighbor distribute-list command](#) [See neighbor distribute list command ]

distribute-list option

[neighbor command](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#)

[dropped field \(show ip bgp neighbors command\)](#)

[Duration field \(show ip bgp flap-statistics command\)](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

## [EBGP \(External BGP\)](#)

[ebgp-multihop option:neighbor command](#)

[options:neighbor command:ebgp-multihop](#) [2nd](#) [3rd](#)

[empty route maps](#)

[effect of](#) [2nd](#)

[Established state](#)

[establishing](#)

[TCP sessions](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[Event Timers Table field \(show ip bgp neighbors command\)](#)

[events](#)

[dampening-related](#)

[debugging](#) [2nd](#)

[neighbor relationship-related](#)

[debugging](#)

[example configuration](#)

[automatic route summarization](#) [2nd](#) [3rd](#)

[example configurations](#)

[synchronization](#) [2nd](#) [3rd](#)

[extended access lists:route filtering](#)

[access lists:extended:route filtering](#)

[external link field \(show ip bgp neighbors command\)](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

fast external failover

[flapping routes:fast external failover](#) [2nd](#)

fields:show ip bgp neighbors command output

output:show ip bgp neighbors command:fields

[output:show ip bgp neighbors command:fields](#) [2nd](#) [3rd](#)

fields:show ip bgp summary command output

output:show ip bgp summary command:fields

[show ip bgp summary command:output:fields;commands:show ip bgp summary:output, fields](#) [2nd](#)

filter-list in option:neighbor command

[options:neighbor command:filter-list in](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#)

filter-list option:neighbor command

[options:neighbor command:filter-list](#) [2nd](#) [3rd](#)

filter-list out option:neighbor command

[options:neighbor command:filter-list out](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

filtering:updates based on AS path information

[updates:filtering based on AS path information](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

[Flags field \(show ip bgp neighbors command\)](#)

flap statistics

[clearing](#)

[Flaps field \(show ip bgp flap-statistics command\)](#)

[For address family field \(show ip bgp neighbors command\)](#)

[Foreign host, Foreign port field \(show ip bgp neighbors command\)](#)

[From field \(show ip bgp flap-statistics command\)](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[\[g\]](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

half-life

[route flap penalties](#)

[hold time field \(show ip bgp neighbors command\)](#)

holdtime:configuring

[keepalive time:configuring](#) [2nd](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[IBGP \(Interior BGP\)](#)

[Idle state](#)

[IGPs \(Interior Gateway Protocols\)](#)

[Inbound path policy field \(show ip bgp neighbors command\)](#)

[incoming route updates:filtering](#)

[filtering:incoming route updates](#)

[updates:incoming:filtering](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[input distribute lists:configuring](#)

[filtering routes:input distribute lists](#)

[configuring:input distribute lists](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[InQ field \(show ip bgp summary command\)](#)

[IP community lists](#)

[irs field \(show ip bgp neighbors command\)](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[keepalive interval field \(show ip bgp neighbors command\)](#)

[keepalive messages:debugging](#)

[messages:keepalive:debugging](#)

[keepalive time:configuring](#)

[holdtime:configuring](#)

[setting:keepalive timers;setting:hold timers](#) [2nd](#)

[keywords](#)

[show ip bgp community command](#)

[known networks](#)

[KRTT field \(show ip bgp neighbors command\)](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[Last read field \(show ip bgp neighbors command\)](#)

[Last reset field \(show ip bgp neighbors command\)](#)

leaking routes to neighbors

[neighbor relationships:routes, leaking](#) [2nd](#) [3rd](#) [4th](#)

learned paths:adding to routing table

[adding:learned paths to routing table](#)

[IP routing table:learned paths:adding;installing:learned routes in IP routing table](#) [2nd](#) [3rd](#)

learned routes:aggregating

[aggregating:learned routes](#)

learned routes:aggregation

[aggregation:learned routes](#) [2nd](#)

learned routes:IBGP:synchronization

[IBGP:synchronization](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

learned routes:limiting

[limiting:learned routes](#)

[prefixes:learned routes:limiting;restricting:learned routes](#) [2nd](#) [3rd](#) [4th](#)

limiting

[output of show commands](#)

[Local host, Local port field \(show ip bgp neighbors command\)](#)

local preference attribute

[attributes:local preference](#)

[local router ID field \(show ip bgp flap-statistics command\)](#)

[local router ID field \(show ip bgp output\)](#)

local routes:aggregating

[aggregation:locally-sourced routes](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[LocPrf field \(show ip bgp output\)](#)

loop detection:configuring

[confiuring:loop detection](#)

[route reflectors:redundancy;redundant route reflectors:configuring;configuring:route reflectors:redu](#)

loopback interfaces

[configuring](#) [2nd](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[main routing table version field \(show ip bgp summary command\)](#)

[manipulating:attributes](#)

[attributes:manipulating](#)

[maximum-paths command](#)

[commands:maximum-paths](#) [2nd](#) [3rd](#)

[maximum-prefix option:neighbor command](#)

[options:neighbor command:maximum-prefix](#) [2nd](#)

[maxRTT field \(show ip bgp neighbors command\)](#)

[MD5 \(Message Digest 5\) authentication:enabling](#)

[authentication:MD5:enabling](#)

[security:MD5 authentication:enabling;connections:TCP:MD5 authentication, enabling](#) [2nd](#)

[MED values:comparing from different ASs](#)

[ASs:MED values:comparing](#)

[comparing:MED values from different ASs;attributes:MED:comparing from different ASs](#)

[memory](#)

[logging neighbor status changes to](#) [2nd](#)

[Metric field \(show ip bgp output\)](#)

[minRTT field \(show ip bgp neighbors command\)](#)

[MsgRcvd field \(show ip bgp summary command\)](#)

[MsgSent field \(show ip bgp summary command\)](#)

[mul-in field \(show ip bgp neighbors command\)](#)

[mul-out field \(show ip bgp neighbors command\)](#)

[multi-exit discriminator \(MED\) attribute](#)

[attributes:multi-exit discriminator \(MED\)](#)

[MED \(multi-exit discriminator\) attribute](#)

[multiple default routes:configuring](#)

[configuring:multiple default routes](#) [2nd](#) [3rd](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

neighbor advertise-map command

[commands:neighbor advertise map](#) [2nd](#) [3rd](#) [4th](#)

neighbor advertisement-interval command

[commands:neighbor advertisement-interval](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[Neighbor capabilities field \(show ip bgp neighbors command\)](#)

neighbor default-originate command

[commands:neighbor default-originate](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

neighbor description command

[commands:neighbor description](#) [2nd](#)

neighbor distribute-list in command

[commands:neighbor distribute-list in](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

neighbor distribute-list out command

[commands:neighbor distribute-list out](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

neighbor ebgp-multihop command

[commands:neighbor ebgp-multihop](#) [2nd](#) [3rd](#)

[Neighbor field \(show ip bgp summary command\)](#)

neighbor filter-list in command

[commands:neighbor filter-list in](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

neighbor filter-list out command

[commands:neighbor filter-list out](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

neighbor filter-list weight command

[commands:neighbor filter-list weight](#) [2nd](#) [3rd](#)

neighbor maximum-prefix warning-only command

[commands:neighbor maximum-prefix warning-only](#) [2nd](#) [3rd](#)

neighbor next-hop-self command

[commands:neighbor next-hop-self](#) [2nd](#) [3rd](#)

neighbor originate route-map command

[commands:neighbor originate route-map](#) [2nd](#) [3rd](#)

neighbor password command

[commands:neighbor password](#) [2nd](#)

neighbor peer-group command

[commands:neighbor peer-group](#) [2nd](#) [3rd](#) [4th](#)

neighbor prefix-list in command

[commands:neighbor prefix-list in](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

neighbor prefix-list out command

[commands:neighbor prefix-list out](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

neighbor relationships:creating

creating:neighbor relationships

[forming:neighbor relationships](#) [2nd](#)

neighbor relationships

[inbound soft reconfiguration, configuring](#)

neighbor relationships:Idle

[Idle neighbor relationships](#)

neighbor relationships:logging changes in

logging:neighbor relationship changes

[syslog:logging neighbor relationship changes;UNIX syslog:logging neighbor relationship changes](#) [2nd](#)

[3rd](#)

neighbor relationships:shutting down

terminating:neighbor relationships

[shutting down:neighbor relationships](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[Open Confirm state](#)

[Open Sent state](#)

[optional attributes](#)

origin attribute

[attributes:origin](#)

[Origin codes field \(show ip bgp flap-statistics command\)](#)

[Origin codes field \(show ip bgp output\)](#)

ouput distribute lists:configuring

filtering routes:output distribute lists

[configuring:ouput distribute lists](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[Outbound path policy field \(show ip bgp neighbors command\)](#)

outgoing route updates:filtering

filtering:outgoing route updates

[updates:outgoing:filtering](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

output distribute lists:configuring

filtering routes:output distribute lists

[configuring:ouput distribute lists](#)

output modifiers

[BGP routing table entries, displaying](#) [2nd](#)

output:show ip bgp command:fields

fields:show ip bgp command output

[show ip bgp command:output:fields;commands:show ip bgp:output, fields](#)

[OutQ field \(show ip bgp summary command\)](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

parameters

[route dampening](#)

password option:neighbor command

[options:neighbor command:password](#) [2nd](#)

[Path field \(show ip bgp flap-statistics command\)](#)

[Path field \(show ip bgp output\)](#)

path selection

[selecting paths](#)

peer group option:neighbor command

[options:neighbor command:peer group](#) [2nd](#)

peer groups:creating

creating:peer groups

[neighbor relationship:peer groups:creating;forming:peer groups](#) [2nd](#) [3rd](#)

peer groups:statistics:displaying

displaying:BGP peer group statistics

[statistics:BGP peer groups:displaying;viewing:BGP peer group statistics](#)

peer groups:verifying configuration

verifying:peer group configuration

[neighbor relationship:peer groups:verifying configuration](#)

penalizing:flapping routes

flapping routes:penalties

[prefixes:route flaps:penalties for](#)

[permitting route advertisement](#)

[Prefix advertised field \(show ip bgp neighbors command\)](#)

prefix-list option:neighbor command

[options:neighbor command:prefix-list](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[prefixes](#) [See also dampened prefixes ]

[advertising](#) [2nd](#)

prefixes:displaying BGP table entry

displaying:prefixes in BGP routing table

[routing table:prefixes, displaying](#)

prefixes:route flaps:dampening

flapping routes:dampening

[dampening:configuring;configuring:route dampening;route dampening:configuring](#)

flapping routes:dampening:default configuration

[dampening:default configuration;configuring:route dampening:default;route dampening:default configur](#)

[2nd](#) [3rd](#) [4th](#)





[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[rcvnext field \(show ip bgp neighbors command\)](#)

[rcvwnd field \(show ip bgp neighbors command\)](#)

[Received field \(show ip bgp neighbors command\)](#)

redistribute protocol command

[commands:redistribute protocol](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

redistribute protocol metric metric command

[commands:redistribute protocol metric metric](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

redistributed routes:aggregating

[aggregating:redistributed routes](#)

redistribution:selective

selective redistribution

[connected networks:selective redistribution;static routes:selective redistribution;learned routes:se](#) [2nd](#)  
[3rd](#) [4th](#) [5th](#)

[regular expressions](#) [2nd](#) [3rd](#)

regular expressions:complex

[complex regular expressions](#)

[remote AS field \(show ip bgp neighbors command\)](#)

[remote router ID field \(show ip bgp neighbors command\)](#)

remote-as option:neighbor command

[options:neighbor command:remote-as](#) [2nd](#) [3rd](#) [4th](#)

remove-private-as option:neighbor command

[options:neighbor command:remove-private-as](#) [2nd](#) [3rd](#)

removing:private AS numbers in updates

updates:private AS numbers, removing

[private AS numbers:removing from updates](#) [2nd](#) [3rd](#)

restricting

[output of show commands](#)

[Reuse field \(show ip bgp flap-statistics command\)](#)

route filtering

[with standard access lists](#) [2nd](#)

route filtering:based on AS path information

[AS path attribute:as route filtering criteria](#) [2nd](#)

route flaps

prefixes:route flaps

[WITHDRAWN messages;messages:WITHDRAWN;UPDATE](#)  
[messages;messages:UPDATE;flapping routes:dampening;damp](#) [2nd](#)

route maps

[AS path filters](#)

empty

[effect of](#) [2nd](#)

route maps:configuring

configuring:route maps

[filtering:routes with route maps](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

route maps:dampening parameters:configuring

dampening:customized parameters:route maps

[flapping routes:dampening:customized parameters](#) [2nd](#) [3rd](#)

route maps:deny/deny form

deny/deny route map form

[logic of route maps:deny/deny form](#) [2nd](#)

route maps:deny/permit form





[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

selecting:best path:based on route attributes

attributes:as best path selection criteria

[best path:selecting:based on route attributes](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#) [12th](#)

selecting:deterministic best path

best path:deterministic selection

[deterministic best path selection](#)

send-community option:neighbor command

[options:neighbor command:send-community](#) [2nd](#) [3rd](#)

[Sent field \(show ip bgp neighbors command\)](#) [2nd](#)

show commands

[commands:show](#) [2nd](#)

show commands:output, reducing

reducing:output from show commands

[output:show commands, reducing](#)

show ip bgp command

[commands:show ip bgp](#)

show ip bgp filter-list command

[commands:show ip bgp filter-list](#)

[show ip bgp flap-statistics](#)

show ip bgp flap-statistics command

[commands:show ip bgp flap-statistics](#) [2nd](#)

[fields:show ip bgp flap-statistics command output:displaying;output:show ip bgp flap-statistics comm](#)

[2nd](#)

show ip bgp inconsistent-as command

[commands:show ip bgp inconsistent-as](#) [2nd](#)

show ip bgp neighbors command

[commands:show ip bgp neighbors](#) [2nd](#) [3rd](#) [4th](#)

show ip bgp neighbors command:output

[commands:show ip bgp neighbors:output](#)

show ip bgp paths command:output, fields

commands:show ip bgp paths:output, fields

[fields:show ip bgp paths command output](#)

show ip bgp summary command

[commands:show ip bgp summary](#) [2nd](#)

show ip route command:output

[commands:show ip route:output](#) [2nd](#)

shutdown option:neighbor command

[options:neighbor command:shutdown](#)

[sndnxt field \(show ip bgp neighbors command\)](#)

[snduna field \(show ip bgp neighbors command\)](#)

[sndwnd field \(show ip bgp neighbors command\)](#)

soft keyword:clear commands

[clear commands:soft keyword](#)

soft-reconfiguration inbound option:neighbor command

[options:neighbor command:soft-reconfiguration inbound](#) [2nd](#)

speakers:full mesh:configuring

full mesh:configuring

[topologies:full mesh:configuring;IBGP:full mesh:configuring](#)

[topologies:full mesh:configuring;IBGP:full mesh:configuring;confederations](#) [2nd](#) [3rd](#)

speakers:states of



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[TblVer field \(show ip bgp summary command\)](#)

TCP sessions:configuring

configuring:TCP sessions

[EBGP:TCP sessions, configuring](#); [IBGP:TCP sessions, configuring](#) [2nd](#) [3rd](#) [4th](#)

timers bgp command

[commands:timers bgp](#) [2nd](#) [3rd](#) [4th](#)

timers option:neighbor command

[options:neighbor command:timers](#) [2nd](#)

[total data bytes field \(show ip bgp neighbors command\)](#) [2nd](#)

transitive optional attributes

[nontransitive optional attributes](#)

troubleshooting

[administrative distance configuration](#)

[BGP default route advertisement](#)

IP routing table

[installed routes](#)

troubleshooting:BGP timer configuration

keepalive timers:troubleshooting

[hold timers:troubleshooting](#)

troubleshooting:synchronization

[synchronization:troubleshooting](#)

[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

[unread input bytes field \(show ip bgp neighbors command\)](#)

unsuppress-map option:neighbor command

[options:neighbor command:unsuppress-map](#) [2nd](#)

[up for field \(show ip bgp neighbors command\)](#)

[Up/Down field \(show ip bgp summary command\)](#)

update-source option:neighbor command

[options:neighbor command:update-source](#) [2nd](#)

updates

[debugging](#)

to/from specific neighbor

[debugging](#) [2nd](#)

updates:debugging

[input updates:debugging](#)

[output updates:debugging](#)

updates:outgoing:filtering based on AS path information

filtering:outgoing updates based on AS path information

[outgoing updates:filtering based on AS path information](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[K](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)]

## [V field \(show ip bgp summary command\)](#)

verifying

[confederation peer relationships](#) [2nd](#)

[installed routes in IP routing table](#) [2nd](#)

redistribution

[of non-BGP learned routes](#) [2nd](#) [3rd](#)

verifying:administrative distance

[administrative distance:verifying](#)

verifying:aggregate address configuration

[aggregation:verifying configuration](#)

verifying:auto-summarization configuration

auto-summarization:configuration:verifying

[configuring:auto-summarization:verification](#)

verifying:BGP timer configuration

keepalive timers:verifying configuration

[hold timers:verifying configuration](#)

verifying:default route configuration

[default routes:verifying configuration](#)

verifying:fast external failover configuration

fast external failover:verifying configuration

[flapping routes:fast external failover:verifying configuration](#)

verifying:synchronization

[synchronization:verifying configuration](#)

version option:neighbor command

[options:neighbor command:version](#) [2nd](#)

viewing:dampened routes

dampened routes:viewing

[displaying:dampened routes](#)



[\[SYMBOL\]](#) [\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#)

weight attribute

[attributes:weight](#)

weight attribute:configuring for local routes

[attributes:weight:configuring for local router](#)

weight attribute:setting

configuring:weight attribute

[attributes:weight:setting](#) [2nd](#) [3rd](#) [4th](#)

[Weight field \(show ip bgp output\)](#)

weight option:neighbor command

[options:neighbor command:weight](#)

[well-known attributes](#)

[optional attributes](#)

[with data field \(show ip bgp neighbors command\)](#) [2nd](#)

[withdrawn field \(show ip bgp neighbors command\)](#)