

# Technology Overview

## Using 4-Byte Autonomous System Numbers in BGP Networks

Release

# 10.4



Published: 2010-10-08

Revision 1

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

#### *Technology Overview Using 4-Byte Autonomous System Numbers in BGP Networks*

Release 10.4

Copyright © 2010, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Miguel Barreiros, Roy Spencer

Editing: Nancy Kurahashi, Roy Spencer

Illustration: Dawn Spencer

Cover Design: Edmonds Design

#### Revision History

October 2010—R1 Junos 10.4

The information in this document is current as of the date listed in the revision history.

#### YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

**READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE.** BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).



# Table of Contents

4-Byte Autonomous System Numbers Overview . . . . .	1
Juniper Networks Implementation of 4-Byte Autonomous System Numbers . . . .	3
Configuring 4-Byte Autonomous System Numbers . . . . .	5
Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number . . . . .	7
Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number . . . . .	9
Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain . . . . .	11
Prepending 4-Byte AS Numbers in an AS Path . . . . .	15
Understanding 4-Byte AS Numbers and Route Distinguishers . . . . .	17
Configuring 4-Byte AS Numbers and BGP Extended Community Attributes . . . .	19
Understanding 4-Byte AS Numbers and Route Loop Detection . . . . .	21





## 4-Byte Autonomous System Numbers Overview

---

This Technology Overview describes 4-byte autonomous system (AS) numbers and the operation of BGP in a network with a mix of 2-byte and 4-byte AS numbers.

The 2-byte AS number, also known as a 16-bit AS number or 2-octet AS number, provides a pool of 65,536 AS numbers. The 2-byte AS number range has been exhausted. 4-byte AS numbers are specified in RFC 4893, *BGP Support for Four-Octet AS Number Space* and provide a pool of 4,294,967,296 AS numbers.

As of January 1, 2009 the Internet Assigned Numbers Authority (IANA) only assigns 4-byte AS numbers, unless a 2-byte AS number is specifically requested. The Internet Engineering Task Force (IETF) RFC 4893 defines a method for smooth transition from 2-byte AS numbers to 4-byte AS numbers and for maintaining backward compatibility.

RFC 4893 introduces two new optional transitive BGP attributes, AS4\_PATH and AS4\_AGGREGATOR. These new attributes are used to propagate 4-byte AS path information across BGP speakers that do not support 4-byte AS numbers.

RFC 4893 also introduces a reserved, well-known, 2-byte AS number, AS 23456. This reserved AS number is called AS\_TRANS in RFC 4893.

This Technology Overview presents the following topics:

- Juniper Networks implementation of 4-byte AS Numbers
- How to configure 4-byte AS numbers
- The impact on BGP peering sessions with a mix of 2-byte and 4-byte AS numbers
- The impact on BGP paths with a mix of 2-byte and 4-byte AS numbers
- How to configure route distinguishers using 4-byte AS numbers
- How to configure extended community attributes using 4-byte AS numbers
- The impact on BGP route loop detection with a mix of 2-byte and 4-byte AS numbers

### Related Documentation

- Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19
- Configuring 4-Byte Autonomous System Numbers on page 5
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3
- Prepending 4-Byte AS Numbers in an AS Path on page 15
- Understanding 4-Byte AS Numbers and Route Distinguishers on page 17
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21

- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11

## Juniper Networks Implementation of 4-Byte Autonomous System Numbers

---

Junos OS Release 9.1 and later supports 4-byte AS numbers.

If your network is currently using 2-byte AS numbers, you are not required to get new 4-byte AS numbers. The 2-byte AS number range is a subset of the 4-byte AS number range. A Juniper networks router that supports 4-byte AS numbers simply prepends a string of zeros in front of the 2-byte AS number. For example, the 2-byte AS number 65000 becomes the 4-byte AS number 00000.65000.

If your Juniper Networks router supports 4-byte AS numbers and has a peer relationship with a router that does not support 4-byte AS numbers, the following sequence takes place in the adjacent RIB-in routing table after the router that supports 4-byte AS numbers advertises this capability to the new peer:

1. The router that supports 4-byte AS numbers receives an advertisement from the peer that supports only 2-byte AS numbers.
2. On the router that supports 4-byte AS numbers, the 2-byte AS path is converted into the 4-byte AS number by prepending a string of zeros in front of the 2-byte AS number.
3. If a 4-byte AS number is also present in the path, it is merged with the 2-byte AS numbers in the path.
4. If the AGGREGATOR and AS4\_AGGREGATOR attributes are present, these attributes are also merged.

If your Juniper Networks router supports 4-byte AS numbers and has a peer relationship with a router that does not support 4-byte AS numbers, the following sequence takes place in the adjacent RIB-out routing table:

1. Update message are reformatted before being sent to the router that does not support 4-byte AS numbers.
2. The router that supports 4-byte AS numbers sends the 4-byte AS number in the AS4\_PATH attribute.
3. The AS\_PATH attribute is also sent. It is encoded with the 2-byte AS numbers. Mappable 4-byte AS numbers, below 64537, are sent as 2-byte AS numbers. Non-mappable 4-byte AS numbers, above 64536, are represented by the well-known 2-byte AS number, AS 23456.
4. A single peer group is used for the routers that support 4-byte AS numbers and the routers that support only 2-byte AS numbers.

### Related Documentation

- 4-Byte Autonomous System Numbers Overview on page 1
- Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19
- Configuring 4-Byte Autonomous System Numbers on page 5
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7

- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Prepending 4-Byte AS Numbers in an AS Path on page 15
- Understanding 4-Byte AS Numbers and Route Distinguishers on page 17
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21
- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11

## Configuring 4-Byte Autonomous System Numbers

---

This section describes how to configure a 4-byte AS number and how to verify if the BGP peer supports 4-byte AS numbers.

The AS number can be specified in plain number format or in AS-dot notation format on routers running Junos OS Release 9.2 and later. For example, the 4-byte AS number of 65,546 is represented in plain-number format as 65546. The same AS number is represented in AS-dot notation format as 1.10 on routers running Junos OS Release 9.2 and later.

- To configure a 4-byte AS number in AS-dot notation format, include the **autonomous-system** statement and specify the 4-byte AS number. In the following example the AS number is set to 1.10.

```
user@host# set routing-options autonomous-system 1.10
```

- To configure a 4-byte AS number in plain number format, include the **autonomous-system** statement and specify the 4-byte AS number. In the following example the AS number is set to 65546.

```
user@host# set routing-options autonomous-system 65546
```

- After a BGP peer session has been negotiated, you can verify whether the peer supports 4-byte AS numbers or not. To verify whether the peer supports 4-byte AS numbers or not, use the **show bgp neighbor** command. In the following example the peer does not support 4-byte AS numbers.

```
user@host# show bgp neighbor 192.168.1.9 | match "AS"
```

```
Peer: 192.168.1.9+179 AS 65056 Local: 192.168.1.3+52616 AS 65000  
Peer does not support 4 byte AS extension
```

- In the following example the peer does support 4-byte AS numbers.

```
user@host# show bgp neighbor 192.168.1.9 | match "AS"
```

```
Peer: 192.168.1.10+52679 AS 1000000000 Local: 192.168.1.3+179 AS 65000  
Peer supports 4 byte AS extension (peer-as 1000000000)
```

### Related Documentation

- 4-Byte Autonomous System Numbers Overview on page 1
- Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3
- Prepending 4-Byte AS Numbers in an AS Path on page 15
- Understanding 4-Byte AS Numbers and Route Distinguishers on page 17
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21

- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11

## Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number

This section describes what happens when a router that supports 4-byte AS numbers establishes a peer relationship with a router that only supports 2-byte AS numbers if both routers are configured with AS numbers in the 2-byte AS number range.

In Figure 1 on page 7, Router 1 is running Junos OS Release 9.2 that supports 4-byte AS numbers. Router 1 is configured to use a 2-byte AS number, AS 12596. Router 2 is running Junos OS Release 8.5 that supports 2-byte AS numbers. Router 2 is configured with a 2-byte AS number, AS 60000.

**Figure 1: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 2-Byte AS Number**



- The following example shows the relevant portion of the Router 1 configuration.

```
user@Router1# show configuration
...
autonomous-system 12596;
...
local-address 192.168.1.10;
export static-to-bgp;
peer-as 60000;
```

- To verify that the AS path of route 1.2.3.4 contains AS 12596, use the **show route** command on Router 2. The following example shows that the BGP peer session is established in the normal way and that the AS path of route 1.2.3.4 contains AS 12596:

```
user@Router2# show route 1.2.3.4

1.2.3.4/32          *[BGP/170] 00:01:29, localpref 100, from 192.168.1.10
                   AS path: 12596 I
                   > via at-0/1/0.1001
```

- To display the session-establishment messages logged on Router 1, use the **show log messages** command. The following example shows that Router 1 discovers that Router 2 does not support 4-byte AS numbers:

```
user@Router1# show log messages

Nov  7 09:41:39.443493 bgp_4byte_aspath_add_cap():153 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): 4 byte AS capability added, AS 12596
Nov  7 09:41:39.443582 bgp_send: sending 67 bytes to 192.168.1.9 (External
AS 60000)
[...]
Nov  7 09:41:39.448055 bgp_4byte_aspath_adjust():1279 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): Adjust BGP update to Old/New BGP speaker format
Nov  7 09:41:39.448132 bgp_4byte_aspath_adjust():1290 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): Cached information of previous update format
```

is not used

Nov 7 09:41:39.448162 bgp\_generate\_2byte\_aspath():422 AS4-Peer 192.168.1.9 (External AS 60000)(SEND): Generating 2 byte AS path from 4 byte as-path

Nov 7 09:41:39.448198 bgp\_send: sending 64 bytes to 192.168.1.9 (External AS 60000)

**Related  
Documentation**

- [4-Byte Autonomous System Numbers Overview on page 1](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19](#)
- [Configuring 4-Byte Autonomous System Numbers on page 5](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 15](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 17](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 21](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11](#)



## Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number

This section describes what happens when a router that supports 4-byte AS numbers establishes a peer relationship with a router that only supports 2-byte AS numbers if the first router is configured with an AS number outside the 2-byte AS number range.

In Figure 2 on page 9, Router 2 is running Junos OS Release 9.2 that supports 4-byte AS numbers. Router 2 is configured to use a 4-byte AS number, AS 1000000. Router 3 is running Junos OS Release 8.5 that supports 2-byte AS numbers. Router 3 is configured with a 2-byte AS number, AS 60000.

**Figure 2: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 4-Byte AS Number**



You can configure a local AS number to be used only during the establishment of the BGP session with a BGP neighbor, but to be hidden in the AS path sent to external BGP peers. To configure the local AS number, include the **local-as** statement, specify the 2-byte AS number to use, 65530, and include the **private** option. With this configuration, only the global AS number, 1000000, is included in the AS path sent to external peers. The following example shows the relevant portion of the Router 1 configuration:

```
user@Router1# show configuration
...
autonomous-system 1000000;
...
local-address 192.168.1.9;
export static-to-bgp;
neighbor 192.168.1.3 {
    peer-as 60000;
    local-as 65530 private;
}
```

The peer AS number on Router 3 should equal the local AS number on Router 1. The following example shows the relevant portion of the Router 3 configuration:

```
user@Router3# show configuration
...
autonomous-system 60000;
...
local-address 192.168.1.3;
neighbor 192.168.1.9 {
    peer-as 65530;
}
```

To verify that the AS path of route 22.1.2.3 contains AS 65530, use the **show route** command on Router 3. The following example shows that the BGP peer session is established and that the AS path of route 22.1.2.3 contains AS 65530:

```
user@Router3# show route 22.1.2.3
```

```
...
22.1.2.3/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.9
                  AS path: 65530 I
                  > via so-1/0/3.0
```

**Related  
Documentation**

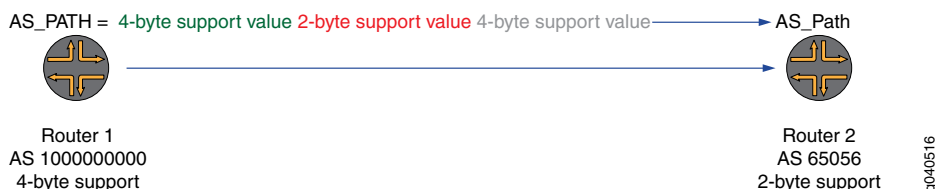
- [4-Byte Autonomous System Numbers Overview on page 1](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19](#)
- [Configuring 4-Byte Autonomous System Numbers on page 5](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 15](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 17](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 21](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11](#)

## Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain

This section describes what happens when a router that supports 4-byte AS numbers sends the AS path statement to a router that only supports 2-byte AS numbers if the first router is configured with an AS number outside the 2-byte AS number range.

In Figure 3 on page 11 Router 1 supports 4-byte AS numbers. Router 1 is configured to use a 4-byte AS number, AS 1000000000. Router 2 supports 2-byte AS numbers. Router 2 is configured with a 2-byte AS number, AS 65056.

**Figure 3: 4-Byte Capable Router AS Path to a 2-Byte Capable Router**



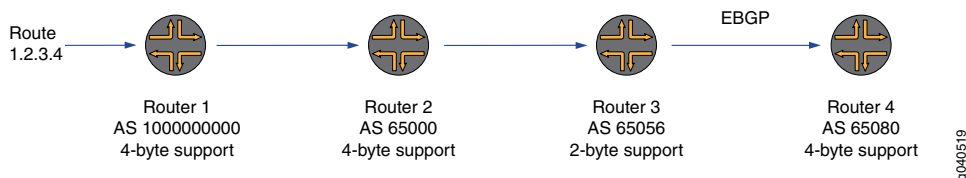
- Router 2 does not accept 4-byte AS numbers in the AS\_PATH attribute. You can verify this using the **show bgp neighbor** command on Router 1.

```
user@Router1# show bgp neighbor 192.168.1.9 | match "AS"
```

```
Peer: 192.168.1.9+179 AS 65056 Local: 192.168.1.2+64053 AS 65080  
Peer does not support 4 byte AS extension
```

Figure 4 on page 11 shows four routers running EBG. Router 1, Router 2, and Router 4 support 4-byte AS numbers. Router 3 does not support 4-byte AS numbers.

**Figure 4: EBG 4-Byte AS Path Through a 2-Byte AS Domain**



In this case:

- Router 1 sends the 4-byte AS number, AS 1000000000, in the AS\_PATH attribute to Router 2.
- Router 2 knows that Router 3 does not support 4-byte AS numbers.
- Router 2 sends the AS\_TRANS number, AS 23456, in the AS\_PATH attribute in place of the 4-byte AS number to Router 3.
- Router 2 sends the 4-byte AS number, AS 1000000000 in the AS4\_PATH attribute to Router 3.

- Because the AS4\_PATH attribute is transitive, Router 3 sends both the AS\_PATH attribute and the AS4\_PATH attribute to Router 4.
- When Router 4 receives the AS\_PATH and AS4\_PATH attributes, it merges the path statements to create an accurate AS path.

You can display the AS path using the **show route** command on Router 3. In the following example, notice that the AS number 23456 appears in the AS path and that the AS4\_PATH attribute is **Unrecognized**. Because the AS4\_PATH attribute is a transitive attribute, it is forwarded to the next router.

```
user@Router3# show route 1.2.3.4 detail
AS path: 65000 23456 I Unrecognized Attributes: 13 bytes
```

You can display the route details using the **show route** command on Router 4. In the following example, notice that as the AS path transitions Router 3, as shown in the AS2 (2-byte AS) path, the AS number is displayed as AS\_TRANS. This means that Router 3 sees the AS number as 23456. In the AS4 (4-byte AS) path the AS number is displayed as 1000000000. In the merged AS path the correct AS path numbers are displayed for AS 65056, AS 65000, and AS 1000000000.

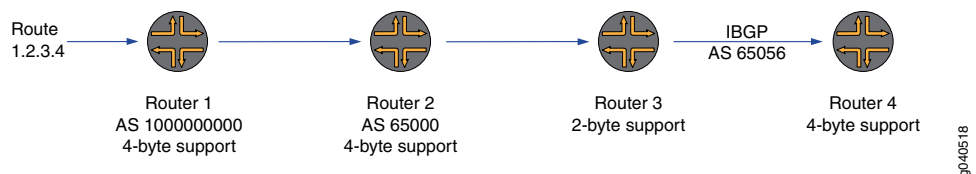
```
user@Router3# show route 1.2.3.4 detail
...
AS path: AS2 PA[3]:65056 65000 AS_TRANS

AS path: AS4 PA[2]:65056 1000000000

AS path: Merged[3]:65056 65000 1000000000 I
```

Figure 5 on page 12 shows 4 routers running IBGP. Router 1, Router 2, and Router 4 support 4-byte AS numbers. Router 3 does not support 4-byte AS numbers.

**Figure 5: IBGP 4-Byte AS Path Through a 2-Byte AS Domain**



In this case:

- Router 1 sends the 4-byte AS number, AS 1000000000, in the AS\_PATH attribute to Router 2.
- Router 2 knows that Router 3 does not support 4-byte AS numbers.
- Router 2 sends the AS\_TRANS number, AS 23456, in the AS\_PATH attribute in place of the 4-byte AS number to Router 3.
- Router 3 sends both the AS\_PATH attribute and the AS4\_PATH attribute to Router 4.
- When Router 4 receives the AS\_PATH and AS4\_PATH attributes, it merges the path statements to create an accurate AS path.

You can display the route details using the **show route** command on Router 2. In the following example, notice that the AS path is displayed as 1000000000.

```
user@Router3# show route 1.2.3.4 detail
```

```
...
```

```
AS path: 1000000000
```

You can display the route details using the **show route** command on Router 3. In the following example, notice that the AS path is displayed as 65000 23456.

```
user@Router3# show route 1.2.3.4 detail
```

```
...
```

```
AS path: 65000 23456 I
```

You can display the route details using the **show route** command on Router 4. In the following example, notice that the merged AS path is displayed as 65000 1000000000.

```
user@Router3# show route 1.2.3.4 detail
```

```
...
```

```
AS path: 65000 1000000000 I
```

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 1](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19](#)
- [Configuring 4-Byte Autonomous System Numbers on page 5](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 15](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 17](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 21](#)

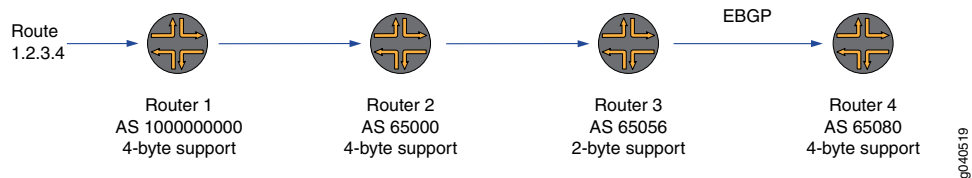


## Prepending 4-Byte AS Numbers in an AS Path

This section describes how to prepend one or more AS numbers at the beginning of an AS path. The AS numbers are added at the beginning of the path after the actual AS number from which the route originates has been added to the path. Prepending an AS path makes a shorter AS path look longer and therefore less preferable to BGP.

In Figure 6 on page 15, Router 2 is configured to prepend AS 1000000000 4 times in front of AS number 65000.

**Figure 6: EBGP With 4-Byte AS Numbers Prepend to the AS Path**



You can display the route details using the `show route` command on Router 3. In the following example, notice that the prepended AS number displayed in the AS path on Router 3 is the AS\_TRANS number, AS 23456. This is because Router 3 does not support 4-byte AS numbers.

```
user@Router3# show route 1.2.3.4 detail
...
1.2.3.4/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.3
                  AS path: 65000 23456 23456 23456 23456 I
```

You can display the route details using the `show route` command on Router 4. In the following example, notice that the prepended AS number displayed in the AS path on Router 4 is AS 1000000000. This is because Router 4 supports 4-byte AS numbers and merges the AS\_PATH and AS4\_PATH attributes.

```
user@Router4# show route 1.2.3.4 detail
...
1.2.3.4/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.9
                  AS path: 65056 65000 1000000000 1000000000 1000000000 1000000000 I
```

### Related Documentation

- 4-Byte Autonomous System Numbers Overview on page 1
- Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19
- Configuring 4-Byte Autonomous System Numbers on page 5
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3
- Understanding 4-Byte AS Numbers and Route Distinguishers on page 17
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21

- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11



## Understanding 4-Byte AS Numbers and Route Distinguishers

---

A route distinguisher (RD) is an 8-byte field prefixed to a service provider customer's IPv4 address. The resulting 12-byte field is a unique VPN-IPv4 address. The RD in BGP messages consists of two major fields, the type field (2 bytes) and value field (6 bytes). The type field determines how the value field should be interpreted.

The route distinguisher is configured as a 6-byte value that you can specify as **as-number:number**, where **as-number** is your assigned AS number and **number** (also known as an administrative number or assigned number subfield) is any 2-byte or 4-byte value. The AS number can be in the range from 1 through 4,294,967,295. If the AS number is a 2-byte value, the administrative number is a 4-byte value. If the AS number is 4-byte value, the administrative number is a 2-byte value.

An RD consisting of a 4-byte AS number and a 2-byte administrative number is defined as a type 2 route distinguisher in RFC 4364, *BGP/MPLS IP Virtual Private Networks*.

To configure an RD using a 4-byte AS number, append the letter "L" to the end of the number. In the following example, the 4-byte AS number is 7765000 and the administrative number is 1000:

```
user@Router1# set routing-instances 4B route-distinguisher 7765000L:1000
```

If the router you are configuring is a BGP peer of a router that does not support 4-byte AS numbers, you also need to configure a local AS number as discussed in "Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number" on page 9. To configure the local AS number, include the **local-as** statement, specify the 2-byte AS number to use (65001), and include the **private** option.

```
user@Router1# set routing-instances 4B protocols bgp group 4B2Bpeers local-as 65001
private
```

### Related Documentation

- 4-Byte Autonomous System Numbers Overview on page 1
- Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19
- Configuring 4-Byte Autonomous System Numbers on page 5
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3
- Prepending 4-Byte AS Numbers in an AS Path on page 15
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21
- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11



## Configuring 4-Byte AS Numbers and BGP Extended Community Attributes

---

A BGP community is a group of destinations that share a common property. You can configure the standard community attribute and extended community attributes for inclusion in BGP update messages.

For example, when configuring a VPN routing and forwarding (VRF) instance, you need to configure a route target. A route target is one type of BGP extended community attribute. To create a named BGP extended community attribute, include the **community** statement and specify the community members:

```
community name {  
    members [ community-ids ];  
}
```

To specify the community members, you must specify the community ID. The community ID consists of three components that you specify in the following format:

*type:administrator:assigned-number*

The **administrator** field of some BGP extended community attributes is an AS number. To configure a **target** extended community, which includes a 4-byte AS number in the plain-number format, append the letter "L" to the end of the number.

In the following example, a **target** community with the 4-byte AS number **334324** and an assigned number of **132** is represented as **target:334324L:132**.

```
[edit policy-options]  
community vpn_blue members [ target:334324L:132 ];
```



NOTE: If you display the target extended community information on a peer router that does not support 4-byte AS numbers, the router displays **target:unknown format**.

### Related Documentation

- 4-Byte Autonomous System Numbers Overview on page 1
- Configuring 4-Byte Autonomous System Numbers on page 5
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7
- Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9
- Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3
- Prepending 4-Byte AS Numbers in an AS Path on page 15
- Understanding 4-Byte AS Numbers and Route Distinguishers on page 17
- Understanding 4-Byte AS Numbers and Route Loop Detection on page 21
- Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11

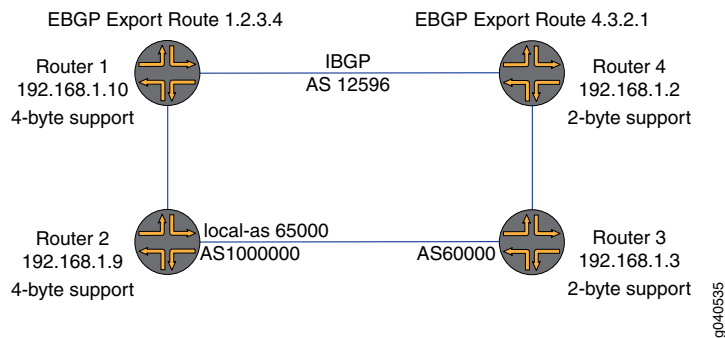


## Understanding 4-Byte AS Numbers and Route Loop Detection

One of the most important functions in BGP is route loop detection at the autonomous system level using the AS\_PATH attribute. A simple way of thinking of the AS\_PATH is that it is the list of autonomous systems that a route goes through to reach its destination. Loops are detected and avoided by the router checking for its own AS number in the AS\_PATH received from a neighboring AS.

This section describes how route loop detection works with a mix of routers that support and do not support 4-byte AS numbers. Figure 7 on page 21 shows a small network with the potential for BGP loops.

Figure 7: 4-Byte AS Numbers and Loop Detection



In the first example, an EBGP route, route 1.2.3.4, is first advertised by Router 1. The first AS in the path is AS 12596 as configured on Router 1. The second AS that is in the path is AS 1000000 as configured on Router 2. AS 1000000 is sent in the AS4\_path attribute and the AS\_TRANS number, AS 23456, is sent in the AS\_PATH attribute to Router 3. The third AS that is in the path is AS 60000, as configured on Router 3.

The **show route** command output shows the AS path for route 1.2.3.4 as advertised by Router 3 to Router 4. In the **show route** command output, you see AS 12596 first. Because Router 3 does not support 4-byte AS numbers, you see AS 23456 second. Because Router 2 used a local AS of 65000 to establish a peer relationship with Router 3, you see AS 65000 third. AS 60000 is not in the **show route** command output because the command was entered on the router configured with AS 60000.

```
user@Router3# show route advertising-protocol bgp 192.168.1.2
...
Prefix Nexthop MED Lc1pref AS path
10.255.14.172/32 Self 65000 23456 12596 I
```

In this case, when Router 4 sees its own AS number, AS 12596, in the path, it detects a routing loop.

In the second example, an EBGP route, route 4.3.2.1, is first advertised by Router 4. The first AS in the path is AS 12596 as configured on Router 4. The second AS in the path is AS 60000 as configured on Router 3. The third AS is in the path is AS 1000000 as configured on Router 2.

The **show route** command output shows the AS path for route 4.3.2.1 as advertised by Router 2 to Router 1. In the **show route** command output, you see AS 12596 first and AS 60000 second. AS 1000000 is not in the **show route** command output because the command was entered on the router configured with AS 1000000.

```
user@Router2# show route advertising-protocol bgp 192.168.1.10
...
Prefix Nexthop MED Lc1pref AS path
10.255.14.172/32 Self 60000 12596 I
```

When Router 1 sees its own AS number, AS 12596, in the path, it detects a routing loop.

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 1](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 19](#)
- [Configuring 4-Byte Autonomous System Numbers on page 5](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 7](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 9](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 3](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 15](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 17](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 11](#)