Important Information

Latest Software
We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

Latest Documentation
The latest version of this document is at:
http://supportcontent.checkpoint.com/documentation_download?ID=13941
For additional technical information, visit the Check Point Support Center (http://supportcenter.checkpoint.com).
For more about this release, see the home page at the Check Point Support Center (http://supportcontent.checkpoint.com/solutions?id=sk67581).

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 April 2012</td>
<td>First release of this document</td>
</tr>
</tbody>
</table>

Feedback
Check Point is engaged in a continuous effort to improve its documentation.
Please help us by sending your comments (mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Advanced Routing Suite CLI R75.40 Reference Guide).
Contents

Important Information ........................................................................................................ 3
The Advanced Routing Suite .............................................................................................. 14
  Overview of the Advanced Routing Suite CLI ............................................................... 14
  About this Guide ............................................................................................................. 14
  Documentation of Commands ....................................................................................... 14
    Overview ....................................................................................................................... 14
    Name ............................................................................................................................ 14
    Syntax ........................................................................................................................... 14
    Mode ............................................................................................................................. 14
    Parameters ................................................................................................................... 14
    Default ........................................................................................................................ 14
    Command History ...................................................................................................... 14
    Examples ..................................................................................................................... 14
    See Also ...................................................................................................................... 14
Using the Advanced Routing Suite CLI ........................................................................... 17
  Starting the Advanced Routing Suite CLI ................................................................. 17
  The -p option ............................................................................................................... 17
  The -f option ............................................................................................................... 18
  The -e option ............................................................................................................... 18
  The -a option ............................................................................................................... 18
  The -s option ............................................................................................................... 18
Basic Features ..................................................................................................................... 19
  Command Tokens ...................................................................................................... 19
  Command Line Completion ......................................................................................... 19
  Moving About the Command Line .............................................................................. 20
  Context-Sensitive Help ............................................................................................... 20
  History .......................................................................................................................... 20
  Disabling/Enabling CLI Tracing ................................................................................... 21
  Aborting an Executing Command .............................................................................. 21
  Screen Paging .............................................................................................................. 21
  Exiting the CLI ............................................................................................................. 21
CLI Modes .......................................................................................................................... 21
  User Execution Mode .................................................................................................. 21
  Privileged Execution Mode .......................................................................................... 22
  Global Configuration Mode .......................................................................................... 22
  Router Configuration Mode .......................................................................................... 22
  Interface Configuration Mode ...................................................................................... 22
CLI Behavior Commands ................................................................................................... 23
  configure file ............................................................................................................... 23
  configure terminal ....................................................................................................... 23
  disable .......................................................................................................................... 23
  enable ........................................................................................................................... 23
  end ................................................................................................................................. 24
  exit ................................................................................................................................. 25
  ip router-id .................................................................................................................. 25
  ip routingtable-id ......................................................................................................... 25
  logout ............................................................................................................................ 25
  quit ................................................................................................................................. 26
  show debugging .......................................................................................................... 26
  show history ................................................................................................................. 28
  show running-config ................................................................................................. 29
bgp cluster-id ................................................................. 84
bgp confederation identifier .................................................. 85
bgp confederation peers ......................................................... 86
bgp non-leading-confeds ....................................................... 86
bgp open-on-accept ............................................................. 88
bgp pass-optional-nontrans .................................................... 88
bgp restart-defer ................................................................. 89
bgp restart-delete-remnants ..................................................... 90
bgp restart-time ................................................................. 91
bgp restart-timeout ............................................................... 91
bgp router-id ..................................................................... 92
bgp send-group-always .......................................................... 93
bgp tie-break-on-age ............................................................. 94
clear ip bgp ..................................................................... 94
default-metric .................................................................. 95
distance ........................................................................ 96
distribute-list ................................................................ 97
enable ........................................................................... 98
maximum-routes ................................................................. 99
neighbor add-communities ..................................................... 100
neighbor aggregator-id .......................................................... 101
neighbor allow .................................................................. 102
neighbor as-loop ................................................................. 103
neighbor as-override ............................................................ 104
neighbor as-path-prepend ....................................................... 105
neighbor capability orf comm-filter ........................................ 105
neighbor capability orf extcomm-filter .................................... 106
neighbor capability orf prefix-filter ....................................... 107
neighbor cluster-id .............................................................. 108
neighbor distance ................................................................. 109
neighbor dynamic ................................................................ 110
neighbor enable ................................................................ 111
neighbor end-of-rib ............................................................... 112
neighbor export-localpref ...................................................... 112
neighbor graceful-restart ....................................................... 113
neighbor ignore-leading-as .................................................... 114
neighbor import-localpref ...................................................... 115
neighbor keep .................................................................. 116
neighbor keepalives-always .................................................... 117
neighbor local-as ................................................................ 118
neighbor log-up-down .......................................................... 119
neighbor maximum-routes ..................................................... 120
neighbor metric-out .............................................................. 122
neighbor multi-protocol-nexthop ............................................. 122
neighbor next-hop-self ........................................................ 123
neighbor orf comm-list ........................................................ 124
neighbor orf extcomm-list ..................................................... 125
neighbor orf prefix-list ........................................................ 126
neighbor out-delay ............................................................... 127
neighbor passive ................................................................ 128
neighbor password ............................................................... 128
neighbor pedantic ............................................................... 129
neighbor peer-group ............................................................. 130
neighbor preference2 ............................................................ 132
neighbor receive-buffer ........................................................ 132
neighbor remote-as ............................................................. 133
neighbor remove-private-as .................................................... 134
neighbor route-map ............................................................. 135
neighbor route-reflector-client .............................................. 136
Fast Open Shortest Path First (OSPF) ........................................ 166
  Fast Open Shortest Path First (OSPF) Overview ................... 168
  router ospf ................................................................. 172
  advertise-subnet .......................................................... 173
  authentication .............................................................. 174
  compatible rfc1583 ......................................................... 176
  dead-interval ............................................................... 176
  distance ..................................................................... 177
  enable ......................................................................... 178
  enable-te .................................................................... 179
  hello-interval ............................................................. 180
  igp-shortcut ............................................................... 181
  inherit-metric ............................................................... 181
  monitor-auth-key ......................................................... 182
  multicast-rib .............................................................. 183
  network area ............................................................... 184
  nssa-inherit-metric ....................................................... 185
  nssa-stability-interval .................................................. 186
  poll-interval .............................................................. 186
  priority ..................................................................... 187
  redistribute ............................................................... 188
  redistribute-nssa .......................................................... 188
  require-vbit ............................................................... 191
  restart-allow-changes ................................................... 192
  restart-enable ............................................................ 193
  restart-max-sync-time ................................................... 194
  restart-type ............................................................... 194
  retransmit-interval ...................................................... 195
  router-id ................................................................. 196
  timers spf ................................................................. 197

Internet Control Message Protocol (ICMP) .......................... 162
  Internet Control Message Protocol (ICMP) Overview ............... 162
  router icmp ............................................................... 162
  trace file ................................................................. 163
  trace flag ................................................................. 164

neighbor route-to-peer .................................................. 137
neighbor send-buffer ...................................................... 138
neighbor send-community ............................................... 139
neighbor soft-reconfiguration inbound .................................. 140
neighbor timers ............................................................ 140
neighbor ttl ................................................................. 142
neighbor update-source ................................................... 142
neighbor use-med .......................................................... 144
neighbor v4-gateway ....................................................... 145
neighbor version ........................................................... 146
network ...................................................................... 146
preference2 ................................................................. 147
redistribute ............................................................... 148
router bgp ................................................................. 150
show ip bgp ............................................................... 151
show ip bgp instance ....................................................... 152
show ip bgp neighbors ..................................................... 152
show ip bgp rfc ............................................................. 154
show ip bgp paths .......................................................... 155
show ip bgp peer-group .................................................... 155
show ip bgp summary ....................................................... 156
timers bgp ................................................................. 157
trace file ................................................................. 158
trace flag ................................................................. 159

router ospf ............................................................... 172
advertise-subnet .......................................................... 173
authentication .............................................................. 174
compatible rfc1583 ......................................................... 176
dead-interval ............................................................... 176
distance ..................................................................... 177
enable ......................................................................... 178
enable-te .................................................................... 179
hello-interval ............................................................. 180
igp-shortcut ............................................................... 181
inherit-metric ............................................................... 181
monitor-auth-key ......................................................... 182
multicast-rib .............................................................. 183
network area ............................................................... 184
nssa-inherit-metric ....................................................... 185
nssa-stability-interval .................................................. 186
poll-interval .............................................................. 186
priority ..................................................................... 187
redistribute ............................................................... 188
redistribute-nssa .......................................................... 188
require-vbit ............................................................... 191
restart-allow-changes ................................................... 192
restart-enable ............................................................ 193
restart-max-sync-time ................................................... 194
restart-type ............................................................... 194
retransmit-interval ...................................................... 195
router-id ................................................................. 196
timers spf ............................................................... 197
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace file</td>
<td>198</td>
</tr>
<tr>
<td>trace flag</td>
<td>199</td>
</tr>
<tr>
<td>transmit-delay</td>
<td>200</td>
</tr>
<tr>
<td>area advertise-subnet</td>
<td>201</td>
</tr>
<tr>
<td>area authentication</td>
<td>202</td>
</tr>
<tr>
<td>area dead-interval</td>
<td>204</td>
</tr>
<tr>
<td>area filter</td>
<td>205</td>
</tr>
<tr>
<td>area hello-interval</td>
<td>206</td>
</tr>
<tr>
<td>area nssa</td>
<td>207</td>
</tr>
<tr>
<td>area nssa-range</td>
<td>208</td>
</tr>
<tr>
<td>area nssa-translate-always</td>
<td>209</td>
</tr>
<tr>
<td>area poll-interval</td>
<td>210</td>
</tr>
<tr>
<td>area priority</td>
<td>211</td>
</tr>
<tr>
<td>area range</td>
<td>212</td>
</tr>
<tr>
<td>area retransmit-interval</td>
<td>213</td>
</tr>
<tr>
<td>area stub</td>
<td>214</td>
</tr>
<tr>
<td>area stubhost</td>
<td>214</td>
</tr>
<tr>
<td>area stubnetwork</td>
<td>214</td>
</tr>
<tr>
<td>area transmit-delay</td>
<td>215</td>
</tr>
<tr>
<td>area virtual-link</td>
<td>216</td>
</tr>
<tr>
<td>default-metric</td>
<td>217</td>
</tr>
<tr>
<td>default-nssa-metric</td>
<td>219</td>
</tr>
<tr>
<td>default-nssa-type</td>
<td>220</td>
</tr>
<tr>
<td>default-preference</td>
<td>221</td>
</tr>
<tr>
<td>default-tag</td>
<td>222</td>
</tr>
<tr>
<td>default-type</td>
<td>222</td>
</tr>
<tr>
<td>advertise-subnet</td>
<td>223</td>
</tr>
<tr>
<td>allow-all</td>
<td>224</td>
</tr>
<tr>
<td>authentication</td>
<td>225</td>
</tr>
<tr>
<td>cost</td>
<td>227</td>
</tr>
<tr>
<td>dead-interval</td>
<td>228</td>
</tr>
<tr>
<td>enable</td>
<td>229</td>
</tr>
<tr>
<td>hello-interval</td>
<td>230</td>
</tr>
<tr>
<td>neighbor</td>
<td>231</td>
</tr>
<tr>
<td>network</td>
<td>232</td>
</tr>
<tr>
<td>no-multicast</td>
<td>232</td>
</tr>
<tr>
<td>passive-interface</td>
<td>233</td>
</tr>
<tr>
<td>poll-interval</td>
<td>234</td>
</tr>
<tr>
<td>priority</td>
<td>235</td>
</tr>
<tr>
<td>retransmit-interval</td>
<td>236</td>
</tr>
<tr>
<td>traffic-eng administrative-weight</td>
<td>237</td>
</tr>
<tr>
<td>traffic-eng attribute-flags</td>
<td>238</td>
</tr>
<tr>
<td>traffic-eng bandwidth</td>
<td>239</td>
</tr>
<tr>
<td>transmit-delay</td>
<td>240</td>
</tr>
<tr>
<td>ip ospf advertise-subnet</td>
<td>241</td>
</tr>
<tr>
<td>ip ospf allow-all</td>
<td>242</td>
</tr>
<tr>
<td>ip ospf area</td>
<td>243</td>
</tr>
<tr>
<td>ip ospf authentication</td>
<td>244</td>
</tr>
<tr>
<td>ip ospf cost</td>
<td>246</td>
</tr>
<tr>
<td>ip ospf dead-interval</td>
<td>246</td>
</tr>
<tr>
<td>ip ospf enable</td>
<td>246</td>
</tr>
<tr>
<td>ip ospf hello-interval</td>
<td>247</td>
</tr>
<tr>
<td>ip ospf neighbor</td>
<td>248</td>
</tr>
<tr>
<td>ip ospf network</td>
<td>249</td>
</tr>
<tr>
<td>ip ospf no-multicast</td>
<td>250</td>
</tr>
<tr>
<td>ip ospf passive-interface</td>
<td>251</td>
</tr>
<tr>
<td>ip ospf poll-interval</td>
<td>251</td>
</tr>
<tr>
<td>ip ospf priority</td>
<td>252</td>
</tr>
<tr>
<td>ip ospf retransmit-interval</td>
<td>254</td>
</tr>
</tbody>
</table>
Routing Information Protocol (RIP) ................................................................. 255
  ip ospf traffic-eng administrative-weight .................................................. 255
  ip ospf traffic-eng attribute-flags .......................................................... 256
  ip ospf traffic-eng bandwidth .................................................................. 257
  ip ospf transmit-delay .............................................................................. 258
  show ip ospf ......................................................................................... 259
  show ip ospf border-routers ..................................................................... 260
  show ip ospf database ............................................................................. 260
  show ip ospf interface ............................................................................ 261
  show ip ospf neighbor ............................................................................ 261
  show ip ospf request-list ......................................................................... 262
  show ip ospf retransmission-list .............................................................. 263
  show ip ospf summary-address .................................................................. 264
  show ip ospf virtual-links ......................................................................... 265

Redirect Processing ......................................................................................... 266
  Redirect Processing Overview ................................................................. 266
  ip redirect ............................................................................................... 266
  router redirect ......................................................................................... 267
  trace file ................................................................................................. 267
  trace flag ............................................................................................... 268

Router Discovery ........................................................................................... 270
  Router Discovery Overview ...................................................................... 270
  ip router-discovery address-policy .......................................................... 271
  ip router-discovery enable ....................................................................... 272
  ip router-discovery trace file .................................................................... 272
  ip router-discovery trace flag ................................................................... 273
  router-discovery lifetime ......................................................................... 274
  router-discovery maximum-interval ......................................................... 275
  router-discovery minimum-interval ......................................................... 276

Routing Information Protocol (RIP) ............................................................... 278
  Routing Information Protocol (RIP) Overview ......................................... 278
  router rip ................................................................................................ 281
  default-metric ........................................................................................ 281
  distribute-list ......................................................................................... 282
  ecmp ....................................................................................................... 285
  enable ..................................................................................................... 285
  flash-update-time ................................................................................... 286
  ignore-host-routes .................................................................................. 287
  ignore-must-be-zero ................................................................................ 287
  network ................................................................................................... 288
  preference ............................................................................................... 289
  query-authentication ............................................................................... 290
  redistribute .............................................................................................. 291
  send-updates ........................................................................................... 293
  source-gateways ..................................................................................... 294
  split-horizon ........................................................................................... 296
    Example 3 ............................................................................................... 297
    term-updates ........................................................................................ 297
    timers basic .......................................................................................... 298
    trace file ............................................................................................... 299
    trace flag .............................................................................................. 300
    trusted-gateways ................................................................................... 301
  ip rip authentication ................................................................................. 301
  ip rip enable ............................................................................................. 302
  ip rip metric-in ....................................................................................... 304
  ip rip metric-out .................................................................................... 305
  ip rip no-receive ..................................................................................... 306
  ip rip no-send ........................................................................................ 307
  ip rip secondary-authentication ............................................................. 308
  ip rip version .......................................................................................... 309
show ip rip database ........................................... 310
Example 3 ..................................................... 311

SNMP Multiplexing (SMUX) ................................ 313
SNMP Multiplexing (SMUX) Overview .................... 313
smux password ................................................. 313
smux port ....................................................... 314
smux trace file ............................................... 315
smux trace flag ............................................... 315

Distance Vector Multicast Routing Protocol (DVMRP) .... 317
Distance Vector Multicast Routing Protocol (DVMRP) Overview .... 317
ip dvmrp ....................................................... 317
ip dvmrp distance ........................................... 318
ip dvmrp default-metric ................................... 319
ip dvmrp disable ............................................. 320
ip dvmrp metric-offset .................................... 320
ip dvmrp nodvmrprout ..................................... 321
ip dvmrp no-retransmit .................................... 322
ip dvmrp prune-lifetime ................................... 323
ip dvmrp trace file ......................................... 323
ip dvmrp trace flag ......................................... 324
ip dvmrp unicast-routing .................................. 326
show ip dvmrp interfaces .................................. 327
show ip dvmrp neighbors ................................... 328
show ip dvmrp route ......................................... 329
tunnel mode dvmrp ........................................... 331

Internet Group Management Protocol (IGMP) .............. 332
Internet Group Management Protocol (IGMP) Overview .......... 332
clear ip igmp group ......................................... 333
ip igmp .......................................................... 334
ip igmp ignore-v1-messages ................................ 335
ip igmp ignore-v2-messages ................................ 335
ip igmp last-member-query-count .......................... 336
ip igmp last-member-query-interval ......................... 337
ip igmp query-interval ....................................... 339
ip igmp query-max-response-time ........................... 340
ip igmp require-router-alert ................................ 341
ip igmp robustness .......................................... 342
ip igmp send-router-alert ................................... 343
ip igmp startup-query-count ................................ 344
ip igmp startup-query-interval .............................. 345
ip igmp static-group ......................................... 347
ip igmp trace file ............................................ 348
ip igmp trace flag ............................................ 349
ip igmp version ............................................... 350
show ip igmp groups ......................................... 351
show ip igmp interface ........................................ 355
show ip igmp interface-summary ............................ 358
show ip igmp static-groups .................................. 359

Protocol Independent Multicast (PIM) .......................... 360
Protocol Independent Multicast (PIM) Overview ............ 360
ip pim assert-holdtime ..................................... 361
ip pim dr-priority ............................................ 362
ip pim hello-holdtime ....................................... 363
ip pim hello-interval ........................................ 363
ip pim ip-holdtime ............................................ 364
ip pim ip-interval ............................................. 365
ip pim lan-delay ............................................... 366
ip pim mr-stale-multiplier ................................. 367
ip pim mrt-interval .......................................... 367
ip pim override-interval ......................................................... 368
ip pim triggered-hello-delay ..................................................... 369
show ip pim control-counters .................................................. 370
show ip pim interface ................................................................ 371
show ip pim neighbor .................................................................. 373

Protocol Independent Multicast - Dense Mode (PIM-DM) ........... 375
  Protocol Independent Multicast - Dense Mode (PIM-DM) Overview...... 375
  ip pim dense-mode ............................................................... 375
  ip pim graft-retry-interval ...................................................... 376
  ip pim require-genid ............................................................. 377
  ip pim source-lifetime ............................................................ 378
  ip pim state-refresh-capable .................................................. 378
  ip pim state-refresh-interval .................................................. 379
  ip pim state-refresh-rate-limit ............................................... 380
  ip pim state-refresh-ttl .......................................................... 381
  ip pim dense trace file .......................................................... 381
  ip pim dense trace flag .......................................................... 382
  show ip pim dense-mode interface-summary ................................ 384
  show ip pim dense-mode mrt ................................................... 384
  show ip pim dense-mode mrt-summary ....................................... 386
  show ip pim grafts .................................................................. 387

Protocol Independent Multicast - Sparse Mode (PIM-SM) .......... 389
  Protocol Independent Multicast - Sparse Mode (PIM-SM) Overview ...... 389
  ip pim associate-msdp ........................................................... 390
  ip pim bsr-admin-scope ......................................................... 390
  ip pim bsr-border ................................................................. 391
  ip pim bsr-candidate ............................................................. 392
  ip pim bsr-candidate global .................................................... 393
  ip pim bsr-candidate group .................................................... 393
  ip pim bsr-candidate interval .................................................. 394
  ip pim bsr-candidate priority .................................................. 395
  ip pim bsr-holdtime ............................................................... 396
  ip pim dr-switch-immediate ..................................................... 396
  ip pim mrt-spt-multiplier ....................................................... 397
  ip pim probe-interval ............................................................. 398
  ip pim register-suppression-timeout ........................................ 399
  ip pim rp-address ................................................................. 399
  ip pim rp-candidate .............................................................. 400
  ip pim rp-candidate advertisement-interval ................................ 401
  ip pim rp-candidate group ..................................................... 402
  ip pim rp-candidate holdtime ............................................... 403
  ip pim rp-candidate priority ................................................... 403
  ip pim rp-switch-immediate .................................................... 404
  ip pim sparse-mode .............................................................. 405
  ip pim threshold ................................................................. 406
  ip pim threshold-dr ............................................................ 407
  ip pim threshold-rp ............................................................. 407
  ip pim trace file ................................................................. 408
  ip pim trace flag .................................................................. 410
  ip pim whole-packet-checksum .............................................. 411
  show ip pim bsr-router ........................................................ 412
  show ip pim cbmr ................................................................. 413
  show ip pim rp ................................................................. 413
  show ip pim rp-candidate ...................................................... 414
  show ip pim rp-hash ............................................................ 414
  show ip pim sparse-mode join-prune xmit .................................. 415
  show ip pim sparse-mode mrt ................................................. 415

Access Lists ............................................................................. 418
  Access Lists Overview ............................................................ 418
match metric-type.................................................................473
match protocol.........................................................................473
match ribs..............................................................................474
match tag ..............................................................................475
route-map ............................................................................476
set as-path prepend.................................................................477
set community-set ................................................................477
set dampen-flap.....................................................................479
set ip next-hop.......................................................................479
set local-preference ................................................................480
set med .................................................................................481
set metric................................................................................482
set metric-type.......................................................................482
set origin..............................................................................483
set preference .......................................................................484
set propagate ........................................................................485
set ribs..................................................................................485
set tag...................................................................................486
Index .......................................................................................489
Chapter 1

The Advanced Routing Suite

In This Chapter

Overview of the Advanced Routing Suite CLI 14
About this Guide 14
Documentation of Commands 14

Overview of the Advanced Routing Suite CLI

The Advanced Routing Suite CLI is available as part of the Advanced Networking Software Blade (http://www.checkpoint.com/products/softwareblades/advanced-networking.html).

For organizations looking to implement scalable, fault-tolerant, secure networks, the Advanced Networking blade enables them to run industry-standard dynamic routing protocols including BGP, OSPF, RIPv1, and RIPv2 on security gateways. OSPF, RIPv1, and RIPv2 enable dynamic routing over a single autonomous system—like a single department, company, or service provider—to avoid network failures. BGP provides dynamic routing support across more complex networks involving multiple autonomous systems—such as when a company uses two service providers or divides a network into multiple areas with different administrators responsible for the performance of each.

Advanced Routing is supported on the Check Point SecurePlatform operating system. For information about SecurePlatform, see the R75.40 SecurePlatform Administration Guide (http://supportcontent.checkpoint.com/solutions?id=sk67581).

The Advanced Routing Suite CLI accepts user entered text commands and sends them to Advanced Routing Suite. These commands can encode a configuration change as well as queries for configuration information and dynamic protocol state.

About this Guide

This guide describes the basic, protocol-independent functionality of the Advanced Routing Suite Command Line Interface (CLI), including command-line completion, logging, and history.

Advanced Routing Suite commands are listed alphabetically within protocol sections. For example, if you are looking for the query-authentication command in RIP, look in the Routing Information Protocol (RIP) (on page 278) chapter, then look for the command under the letter A. You can also use the Index to quickly search for a command.

Documentation of Commands

Most chapters in this guide have these sections:

- Overview of ... (one per chapter)
- Name
- Syntax
- Mode
- Parameters
- Description
Overview

Each chapter includes an Overview section. In most cases, this section describes a protocol or policy. Unlike the remaining sections, each chapter includes only one Overview section.

Name

The Name section lists the name and a short description of the command. For example, the `key` command in RIP:

```
key - sets a RIP MD5 key
```

Syntax

The Syntax section lists the valid syntax configuration, including the `no` configuration (where applicable). For example, configure the IGMP robustness to be 4 using the following syntax:

```
ip igmp robustness 4
```

Notation for parameters

In this manual, the allowed values for each parameter are listed similar to below:

Parameter: `[ max-size size [ k | m ] ]`?

Parameter: `address-family [ ipv4 | ipv6 ] {0,2}`

The words in italics are user-entered commands that must be typed exactly as shown. The words in italics give a type of value. Some common types are `size`, `time`, or `interface-name`.

A pipe in a syntax `|` separates alternatives: one of them must occur. A double pipe `A || B` means that either A or B or both must occur, in any order. Brackets `{}` are for grouping. Juxtaposition is stronger than the double bar, and the double bar is stronger than the bar. Thus "a b | c || d e" is equivalent to "[ a b ] || [ c || [ d e ]]."

A pair of numbers in curly braces `{A,B}` indicates that the preceding type, word or group is repeated at least A and at most B times.

**Note:** A question mark `?` indicates that the preceding type, word or group is optional.

Therefore, in the preceding example, specifying a `max-size` is optional. However, if you do specify a `max-size`, you must enter a value for the `size` and specify either `k` or `m`.

Mode

The Mode section shows the modes in which the command is valid. Some commands are valid in multiple modes. For those, the Description section details how the affects of those configurations differ in Advanced Routing Suite.

Parameters

The Parameters section lists the information that is accepted in the referenced configuration. It includes a description of what sort of parameter Advanced Routing Suite expects (for example, the number of seconds for a query), and the range of values Advanced Routing Suite expects. (For example, the startup-query interval in IGMP accepts a value between 0 and 31744.)
**Note:** If the parameter is a value that is user-defined, such as a time or a name, then the parameter is displayed in italics (for example, *time* or *value*). If the parameter is one of several predetermined options, such as version 1, 2, or 3 in IGMP, then that parameter is displayed in bold courier new format (for example, *version 3*).

**Description**

The Description section includes a detailed description of the configuration.

**Default**

The Default section includes the default value(s) of the command and its content.

**Command History**

The Command History section indicates when the command was first introduced. It can also indicate whether the command, its defaults, or any of its parameters have changed.

**Examples**

The Examples section lists valid configurations for a specified command.

**See Also**

Some commands will include a relevant See Also section. The See Also section lists other commands or sections of this guide that might be useful. In addition, other publicly available documents, such as RFCs, may be listed here.
Chapter 2

Using the Advanced Routing Suite CLI

Starting the Advanced Routing Suite CLI

Be sure no other users are connected to Advanced Routing Suite. With Advanced Routing Suite installed properly and running:

1. Enter the SecurePlatform expert mode.
2. Type pro enable at the prompt and press Enter.
3. Reboot.
4. Type router at the prompt and press Enter.
This begins your CLI session in User Execution mode.

The Advanced Routing Suite CLI can be started with several additional command line options. These options include the following:

- `-p`
- `-f`
- `-l`
- `-a`
- `-s`

**Note** - If the CLI is given an invalid command line option, then it prints out a list of valid options, arguments for those options with a short description of each, then exits without connecting to Advanced Routing Suite.

**The -p option**

```
-p <port>
```

The -p option specifies the port on which Advanced Routing Suite's XML subsystem is listening. The argument to this option must be a valid port number. If the -p option is not specified, then the CLI assumes that Advanced Routing Suite's XML subsystem is listening on port 4242.
The **-f option**

`-f <cmd_fname>`

The `-f` option specifies a file from which CLI commands are to be read after the CLI has initialized.

The **-e option**

`-e [ ( s || c || e || w || d || i ) | 0 | all ]`

The `-e` option specifies the event class or classes that the CLI will log. The arguments have the following meanings:

- `s` - Trace security events
- `c` - Trace user-typed commands
- `e` - Trace errors
- `w` - Trace internal warnings and errors
- `d` - Trace debugging events
- `i` - Trace informational events
- `0` - Trace no events
- `all` - Trace all event sets

**Default**

If the `-e` option is not specified, it is the same as if `-e wg` were specified.

The **-a option**

`-a [ o | a | m ]`

The `-a` option specifies the action to take if you want a log file, and a cli.log file already exists in the location specified by the `-i` option. Available arguments include the following:

- `o` - Overwrite the existing cli.log file
- `a` - Append to the existing cli.log file
- `m` - Move the existing cli.log file to cli.log.<x>, where <x> is the next highest integer among the other files named cli.log.* in the logging directory.

**Default**

If `-a` is not specified, then it is as if `-a a` were specified.

The **-s option**

`-s <size_num>[k | m]`

The `-s` option specifies the maximum size of the CLI log in either kilobytes or megabytes. The action to take when the current log's maximum size is reached is determined by the `-a` option.

**Default**

If the `-s` option is not specified, then it is as if `-s 5m` were specified.
Basic Features

Basic features of the Advanced Routing Suite CLI include the following:

- Command Tokens
- Command Line Completion
- Moving About the Command Line
- Context-Sensitive Help
- Command History
- Disabling/Enabling CLI Logging
- Aborting an Executing Command
- Exiting the CLI

Command Tokens

The Advanced Routing Suite CLI command strings are composed of space-delimited tokens. The maximum number of tokens permitted per line is 32. After a full command line is typed, the Enter key sends the line to the CLI for processing. The CLI is case insensitive.

Command Line Completion

The max number of characters per line is 1024. At any point when typing a command line, you can hit the Tab key to either complete the current command token or show a list of possible completions. Consider the following command structure as an example:

```
   abc
       |   -------
       |       |  
       bar   groove
       |   |  
       -------
       |   |  
       par-name1 par-name2
       |   |  
       number number
```

The valid complete command strings are the following:

```
abc groove
abc bar par-name1 [number]
abc bar par-name2 [number]
```

With command line completion, when you type

```
ab<Tab>
```

the command will be completed as `abc` on the same line because no other legal token begins with "ab".

When you type

```
abc bar <Tab>
```

the CLI will display the tokens that can follow `bar` on a separate line, then re-display your typed line as shown below. Note that "routerz" is the Advanced Routing Suite CLI prompt, with "routerz" being the name of the machine on which Advanced Routing Suite is running.

```
routerz> abc bar <Tab>
par-name1 par-name2
routerz> abc bar
```
Note - The space between "bar" and <Tab> is required for the legal token list to display.

Valid commands are not required to be composed of complete tokens. Only a token's smallest unique abbreviation is required. For example, the following two command strings are equivalent:

```
abc bar par-name1 20
a b par-name1 20
```

If the abbreviation is not unique, the CLI will respond with an "Invalid command" error.

### Moving About the Command Line

The cursor does not need to be at the end of a command line before hitting the Enter key. In the examples below, the underscore indicates the position of the cursor.

```
routerz> abc bat bas_
```

If, in the example above, you intended to type "abc bar bas", move the left arrow key back to the space following "bat", delete the "t", and type "r".

```
routerz> abc bar_bar
```

With the cursor still just right of the "r", you can still hit the ENTER key, and the complete line will be given to the CLI for processing.

### Context-Sensitive Help

Type "?" immediately after any token to obtain context-sensitive help about the last command that you typed. For example requesting help immediately after typing "router" shows you that the command enters router mode:

```
(config)# router?
router Enter router mode
```

Type "?" followed by a space after any set of tokens to obtain a list of options that can be used in the command. For example, if you type the following:

```
(config)# router ?
```

the CLI will respond with the following:

```
aggregate Configure Aggregate/Generate routes
bgp Configure BGP
icmp Configure ICMP
ospf Configure OSPF
rip Configure RIP
```

If "router" was not a valid sequence of tokens (or, if it was misspelled), then the CLI would respond with an "Error completing word" error.

Note - Because the "?" special character is used for Help, it cannot be included in any character string. In other words, a "?" cannot be used when configuring a route map name, a prefix list name, and so on. Doing so will display Help for the command, as shown in the example below.

```
(config)# access-list an?
```

<name> Name of an access list

### History

All commands entered during a CLI session are saved in a command history. The command history can be viewed and with short-hand "!" commands. The history can be toggled on and off.
Disabling/Enabling CLI Tracing

The CLI provides a flexible tracing mechanism. Events to be traced are divided into several classes, each of which can be traced individually. Classes can be traced to any or all three of the following locations: the terminal, a file, or the underlying system's tracing system (i.e., syslog).

Aborting an Executing Command

It may sometimes be desirable to abort a query that generates a lot of output. Typing Ctrl+C generates such an abort signal and flushes any queued input.

Screen Paging

If a response to a command contains more lines than provided by the command line window, then the word "more" appears at the bottom of the screen to indicate that not all lines have been displayed. Press the Space bar to display more lines. To stop viewing the output and return to the command line prompt, press any other key.

Exiting the CLI

Changes are saved as soon as you hit "Enter" after a command. Use the "quit" command to exit the CLI.

CLI Modes

The Advanced Routing Suite CLI has the following five modes:

- User Execution
- Privileged Execution
- Global Configuration
- Router Configuration
- Interface Configuration

The current mode is easily discerned by examining the current command line prompt. The CLI prompt always indicates the current mode. The modes and prompts are described in the sections that follow.

User Execution Mode

User Execution mode is the default mode that the CLI assumes when it begins execution. In User Execution mode, the prompt is ":>".

Note - If the CLI is started with the "-f <cmd_fname>" parameter (see The -f option (on page 18) for more information), then the commands contained in cmd_fname could leave the CLI in something other than User Execution mode when command-line entry control is turned over to the user.

Within User Execution mode, the following actions are allowed:

- Querying of Advanced Routing Suite configuration state
- Querying of dynamic protocol state (for example, the number of OSPF neighbors)
- Modification of various CLI options, such as command history length, CLI events to trace, and so on.
Privileged Execution Mode

Privileged Execution mode allows for "privileged" commands. In Privileged Execution mode, the prompt is "+". This mode is password protected and is entered using enable as follows:

```
routerz> enable
Password: [password]
routerz#
```

Note - The CLI allows three attempts at the "Password:" prompt before returning to the "+" prompt.

Use the disable command to leave Privileged Execution mode and return to User Execution mode.

Global Configuration Mode

Global Configuration mode is used to change the configuration of Advanced Routing Suite. From this mode, you can stop and start protocols and set protocol-specific parameters. This mode can only be entered from Privileged Execution mode with the configure terminal command. When this mode is entered, the prompt changes to "(config)#" as shown below.

```
routerz> enable
Password: [password]
routerz# configure terminal
routerz(config)#
```

Unless otherwise noted, configuration changes that are entered in this mode are made immediately upon hitting Enter.

To exit Global Configuration mode and return to Privileged Execution mode, use the "exit" or "end" command or type "Ctrl+Z". All three are synonymous.

```
routerz(config)# end
routerz#
```

Router Configuration Mode

Router Configuration mode is used to change the protocol state on a specific router. This mode is entered by typing the following at the (config)# prompt:

```
router protocol_name
```

For example, type the following to enter Router Configuration mode for the ICMP protocol:

```
routerz(config)# router icmp
routerz(config-icmp)#
```

The prompt changes to "(config-[protocol_name])#" in Router Configuration mode. To exit Router Configuration mode and return to Global Configuration mode, use the "exit" command.

```
routerz(config-icmp)# exit
routerz(config)#
```

Interface Configuration Mode

Interface Configuration mode is used to change protocol state on a specific interface. This mode is entered by typing the following at the (config)# prompt:

```
interface [ if-type if-number | if-name ]
```

For example, type the following to enter Interface Configuration mode for the physical interface named ppp-interface-0:

```
routerz(config)# interface ppp-interface-0
routerz(config-if)#
```

The prompt changes to "(config-if)#" in Interface Configuration mode. To exit Interface Configuration mode and return to Global Configuration mode, use the "exit" command.
CLI Behavior Commands

The section describes the commands that control the CLI behavior, as opposed to commands that control Advanced Routing Suite behavior.

configure file

Syntax

configure file [filename | replace filename]

Mode

Privileged Execution

Parameters

filename - the name and/or path of the configure file
replace filename - specifies to replace the current configuration file with the specified filename

Description

The configure file command is used to enter an atomic batch mode, where configuration commands are read from the named file or replace with the named file. If any errors are encountered during processing of the named file, then the router's configuration is left unchanged. The filename argument to file can be either a fully or partially qualified name. A fully qualified file name begins with "/" and gives the complete path to the file in addition to the file name. A partially qualified file name does not begin with"/" and may indicate path information in addition to the file name. If path information is given, it is interpreted with respect to the CLI's working directory.

Examples

In the following example, configuration commands are read from the file, /etc/routerz.cfg.

routerz# configure file /etc/routerz.cfg
routerz#

configure terminal

Syntax

configure terminal

Mode

Privileged Execution Mode

Description

Use the configure terminal command in Privileged Execution mode to enter Global Configuration mode and change the router's configuration. This command takes one of two parameters. The "#" prompt changes to "(config)#" to indicate the changed mode. Use the "end" command to leave Global Configuration mode.
Examples
In the following example, configuration commands are entered from the terminal.

```
routerz# configure terminal
routerz(config)# terminal history size 1024
routerz(config)# end
routerz#
```

disable

Syntax
disable

Mode
Privileged Execution

Description
Use the disable command to leave Privileged Execution Mode and re-enter User Execution mode.

Examples
```
routerz# disable
routerz>
```

enable

Syntax
enable

Mode
User Execution

Description
Use the enable command to enter Privileged Execution mode from User Execution mode. The # prompt indicates that the current mode is Privileged Execution mode.

Note - A password is required to enter Privileged Execution mode.

Examples
```
routerz> enable
Password: abcdefg
routerz#
```
**end**

**Syntax**

end

**Mode**

Global Configuration

**Description**

Use the `end` command to leave Global Configuration mode and return to Privileged Execution mode.

**Examples**

```
routerz(config)# end
routerz#
```

**exit**

**Syntax**

exit

**Mode**

Interface Configuration

Global Configuration

**Description**

Use the `exit` command to leave Interface Configuration mode and return to Global Configuration mode. Or, use `exit` to leave Global Configuration mode and return to Privileged Execution mode. Finally, use `exit` to leave Router Configuration mode and return to Global Configuration mode.

**Examples**

```
routerz(config-if)# exit
routerz(config)#
```

or

```
routerz(config)# exit
routerz#
```

**ip router-id**

**Syntax**

ip router-id ipv4 address

**Mode**

Global Configuration
Parameters
ipv4_address - a valid IPv4 address

Description
Use the ip router-id command to configure the global Router ID. This router ID can be overridden in BGP, OSPF, and OSPF3.

Examples
routerz(config)# ip router-id 1.1.1.1
routerz(config)#

ip routingtable-id

Syntax
ip routingtable-id id

Mode
Global Configuration

Parameters
id - an integer from 0 to 250, inclusive

Description
Use the ip routingtable-id command to configure an ID for the routing table. When there is no kernel VRF support, this ID is used to decide the kernel routing table to which Advanced Routing Suite will write.

Examples
routerz(config)# ip routingtable-id 200
routerz(config)#

logout

Syntax
logout

Mode
User Execution

Description
Use the logout command to exit the CLI. This command is synonymous with exit.

Examples
The following example logs you out of the CLI.
Note - The "%" prompt is intended to indicate the shell command line prompt. The prompt can be different, depending on the shell and shell settings from which the Advanced Routing Suite CLI was started.

```
routerz> logout
%
```

### quit

**Syntax**

```
quit
```

**Mode**

User Execution

**Description**

Use the `quit` command to exit the CLI. This command is synonymous with `logout`.

**Examples**

The following example exits you out of the CLI. Note that the "%" prompt is intended to indicate the shell command line prompt. The prompt can be different, depending on the shell and shell settings from which the Advanced Routing Suite CLI was started.

```
routerz> quit
%
```

### show debugging

**Syntax**

```
show debugging
```

**Mode**

User Execution

Privileged Execution

Global Configuration

Interface Configuration

**Description**

This command is used to report information that is of use mainly to developers.

**Examples**

```
routerz> show debugging
...debugging output...
routerz>
```
**show history**

**Syntax**

`show history`

**Mode**

User Execution  
Privileged Execution  
Global Configuration  
Interface Configuration  

**Description**

Use `show history` in any mode to display the commands that have been entered during the current session. The format of the history lines is as follows:

```
command number   time-stamp   command
```

- `command number` represents the sequential number of the command. The most recent command displays with the highest number.  
- `time-stamp` displays the time when the command was entered. Finally, `command` shows the command that was entered.

To re-run the most recent command, type `!!`, and press Enter.

To re-run a specific command appearing in the history list, type `!<number>`, where `<number>` is the command number as it appears in the output of a `show history` command.

**Examples**

**Example 1**

In the following example, several commands are entered. The command history is then displayed.

```
routerz> enable  
Password: abcdefg  
routerz# configure terminal  
routerz(config)# interface fxp0  
routerz(config-if)# ip igmp robustness 2  
routerz(config-if)# show history  
0 09:15:02   enable  
1 09:15:48   configure terminal  
2 09:16:12   interface fxp0  
3 09:17:03   ip igmp robustness 2  
4 09:17:50   show history  
routerz(config-if)#
```

**Example 2**

In the following example, several commands are entered, followed by a `show history` command. Finally, a short-hand `!` command is used to re-run a previous command.

```
routerz> enable  
Password: abcdefg  
routerz# configure terminal  
routerz(config)# interface fxp0  
routerz(config-if)# ip rip metric-in 2  
routerz(config-if)# Ctrl+z  
routerz# show history  
0 10:24:10   enable  
1 10:25:17   configure terminal  
2 10:26:48   interface fxp0
```
**show running-config**

**Syntax**

`show running-config`

**Mode**

Privileged Execution

**Description**

The `show running-config` command reports the active configuration.

**Examples**

The following example shows output for a request for all configuration information.

```
routerz# show running-config
Building configuration...

Current configuration : 966 bytes
!
version 12.2
service password-encryption
!
hostname router-abc
!
enable password 7 011204070A59554E
!
ip subnet-zero
!!
ip multicast-routing
!
interface Ethernet0/0
ip address 10.139.10.26 255.128.0.0
ip pim bsr-border
ip pim sparse-mode
half-duplex
!
interface Serial0/0
ip address 10.11.41.21 255.255.255.252
!
interface Ethernet0/1
ip address 10.11.111.26 255.255.255.0
ip pim sparse-mode
!
router ospf 7
log-adjacency-changes
!
router bgp 7
bgp log-neighbor-changes
!
ip default-gateway 10.254.10.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.254.10.1
no ip http server
```
ip ospf name-pookup

access-list 1 permit 224.0.0.0 15.255.255.255
 !end

routerz#

**show version**

**Syntax**

show version

**Mode**

User Execution
Privileged Execution

**Description**

This command displays the version of the Advanced Routing Suite CLI as well as the version of Advanced Routing Suite.

**Examples**

The following example is a request to determine the version of Advanced Routing Suite currently running.

routerz> show version
CLI version: 1.1
GateD version: ngc 2.3
AMI API version: 2
routerz>

**terminal history**

**Syntax**

terminal history
terminal no history

**Mode**

User Execution
Privileged Execution
Global Configuration
Interface Configuration

**Description**

The `terminal history` command can be used in any mode to turn on command line history. The `terminal no history` command turns off command line history.
Examples

In the following example, command line history is turned on with the first command. Several commands are then entered, followed by a command to turn the command line history off.

```
routerz> terminal history
routerz> configure terminal
routerz(config)# interface 10.10.11.1
routerz(config-if)# ip dvmrp metric-offset in 2
routerz(config-if)# exit
routerz(config)# interface fxp0
routerz(config-if)# ip rip metric-in 1
routerz(config-if)# exit
routerz(config)# terminal no history
```

**terminal history size**

**Syntax**

```
terminal history size num-lines
```

**Mode**

- User Execution
- Privileged Execution
- Global Configuration
- Interface Configuration

**Description**

Use the `terminal history size` command in any mode to set the maximum number of lines that are remembered in the command line history. The default size of the command line history buffer is 1000 lines. The valid range is from 0 to 10,000, inclusive.

**Examples**

The following example sets the command line history buffer to 10 lines.

```
routerz> terminal history size 10
```

**terminal length**

**Syntax**

```
terminal length num-lines
```

**Mode**

- User Execution
- Privileged Execution
- Global Configuration
- Interface Configuration
Description

Use the `terminal length` command in any mode to specify the number of lines that display on a page. The number of lines defaults to the length of your screen. The valid range for `length` is 0 to 4,294,967,295, inclusive.

Note

- Specifying 0 for `length` sets the terminal length to infinity
- If the number of lines specified is less than the number of lines to be displayed, the output for the remaining lines will display in the interactive pager similar to the following:

```
-- MORE (<space> = next page; <enter> = next line; <Q> = stop) --
```

Examples

The following example sets the number of displayed lines to 40 lines.

```
routerz> terminal length 40
```

`write memory`

Syntax

`write memory`

Mode

Privileged Execution

Description

Use the `write memory` command in Privileged Execution mode to write the current Advanced Routing Suite config file for use at startup.

Examples

The following example configures `write memory`.

```
routerz# write memory
```

Querying the Advanced Routing Suite CLI

Use the Advanced Routing Suite queries to request information about a CLI session. Both the candidate and the committed configurations can be queried at any time during a CLI session. Unless otherwise specified, queries can be issued for the entire configuration hierarchy or a subtree of the hierarchy. Protocol-specific query information is available in each chapter. Queries can also be performed to determine memory and task information.

Memory Information

Use the `show memory` query to obtain information about Advanced Routing Suite's current memory usage.
Syntax

```
show memory
```

Parameters

none

Description

The `show memory` query displays information about Advanced Routing Suite's current memory usage.

Examples

The following example displays a response for the `show memory` query.

```
> show memory
Block Name: "nospf_if_head_t"
Block Size: 4
Freelist Length: 1005
Num Init Requests: 1
Num Alloc Requests: 0
Num Free Requests: 0
Num blocks in use: 0
Num bytes in use: 0
Total bytes consumed: 56516
Total bytes in use: 159692
Num block alloc calls: 53767
Min used block size: 4
Page size: 4096
Num of pages allocated: 4096
Num task block malloc pages: 155
Num task block alloc pages: 28
Multipage max alloc: 41
Multipage max reused: 86
Task page alloc multiq: 5
Task block reclaim shreds: 0
Task block reclaim unmaps: (null)
Num multipage pages in use: 95
Num pool alloc pages: 0
Growable array information:
Num of growable arrays: 33
Num of growths: 7
Max allocation: 256
Num malloc calls: 1400
Num calloc calls: 7
Num reallocs: 2532
Num reallocs for more: 11462
Num reallocs for less: 27
Num reallocs for same: 8
Num free-calls: 8
Num bytes requested: 0
Num bytes allocated: 0
Num bytes wasted: 11240
Max outstanding allocs: 41932
Max request: 59848
Num outstanding allocs: 17632
```

Field Descriptions

The following table describes the fields that appear in the Show Memory Query.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Name</td>
<td>The name associated with this block allocator.</td>
</tr>
<tr>
<td>Block Size</td>
<td>This value shows the size, in bytes, of each block allocated by the block memory allocator. Its value is a non-negative integer.</td>
</tr>
<tr>
<td>Freelist Length</td>
<td>This value indicates the length of the freelist associated with the block memory allocator. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num Init Requests</td>
<td>This value shows the number of times that a block memory allocator has been initialized. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num Alloc Requests</td>
<td>This value indicates the number of block memory allocation requests that a block memory allocator has received. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num Free Requests</td>
<td>This value indicates the number of block memory free requests that a block memory allocator has received. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num blocks in use</td>
<td>This value indicates the number of memory blocks that are currently outstanding (or in use) for a memory block allocator. Its value is a non-negative integer.</td>
</tr>
<tr>
<td>Num bytes in use</td>
<td>This value indicates the number of bytes that are currently outstanding (or in use) for a block memory allocator. This value is equal to the value of block-size multiplied by the value of num-blocks-in-use and is always a non-negative integer.</td>
</tr>
<tr>
<td>Total bytes consumed</td>
<td>This value indicates the number of bytes that are associated with all block memory allocators, either in use or freed. Its value is a non-negative integer.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Total bytes in use</td>
<td>This value indicates the number of bytes that are outstanding from all block memory allocators. This is different from the value of total-bytes-consumed, which represents the number of bytes in use or freed summed over all block memory allocators.</td>
</tr>
<tr>
<td>Num block alloc calls</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Min used block size</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Page size</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num of pages allocated</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num task block malloc pages</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num task block alloc pages</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Multipage max alloc</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Multipage max reused</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task page alloc multiq</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task block reclaim shreds</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task block reclaim unmaps</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num multipage pages in use</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num pool alloc pages</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Growable array information</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num of growable arrays</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num of growths</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Max allocation</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num malloc calls</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num calloc calls</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num reallocs</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num reallocs for more</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num reallocs for less</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num reallocs for same</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num free-calls</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num bytes requested</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num bytes allocated</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Num bytes wasted</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Max outstanding allocs</td>
<td>This value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
Max request | This value is a non-negative integer. This information is useful mostly to developers.
Num outstanding allocs | This value is a non-negative integer. This information is useful mostly to developers.

**Task Information**

Use the `show task` query to obtain information about currently active Advanced Routing Suite tasks.

**Syntax**

```plaintext
show task
```

**Parameters**

none

**Description**

The `show task` query displays information about currently active Advanced Routing Suite tasks.

*Note* - Obtaining information about a specific task is not supported.

**Examples**

The following example displays a response for the `show task` query.

```plaintext
> show task
Task Name: "IF"
Task Proto Number: N/A
Task Priority: 10
Task Address: N/A
Task Port: N/A
Task Socket: N/A
Task RT Proto Bit: Direct
Task Flags:
  Accept: no
  Connect: no
  Delete: no
  Low-Priority: no
```

**Field Descriptions**

The following table describes the fields that appear in the Show Memory Information Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>The name of the task in double quotes</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Task Proto Number</td>
<td>The Advanced Routing Suite internal protocol number. This number has no direct correspondence to any type of protocol number carried in a data packet. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task Priority</td>
<td>Shows the priority of the Advanced Routing Suite task. This priority is used to schedule various task-specific operations. Its value is a non-negative integer. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task Address</td>
<td>The ipv4 address that is associated with this task</td>
</tr>
<tr>
<td>Task Port</td>
<td>The port number associated with this task. Typically, only tasks associated with connection-oriented protocols will have a port number of 0 or greater. For all other tasks, the value of this tag is -1.</td>
</tr>
<tr>
<td>Task Socket</td>
<td>The socket number associated with the task. Typically, only tasks associated with connection-oriented protocols will have a port number of 0 or greater. For all other tasks, the value of this tag is -1.</td>
</tr>
<tr>
<td>Task RT Proto Bit</td>
<td>The task's RTRPROTO bit. This name is always contained in double quotes. This information is useful mostly to developers.</td>
</tr>
<tr>
<td>Task Flags</td>
<td>A field for all flags associated with a task</td>
</tr>
<tr>
<td>Accept</td>
<td>Shows the setting for the TASKF_ACCEPT bit for the indicated task. This bit is set if the task is accepting incoming connections. There are some tasks for which this bit is never set. This flag is useful mostly to developers.</td>
</tr>
<tr>
<td>Connect</td>
<td>Shows the setting for the TASKF_CONNECT bit for the indicated task. This bit is set if the task's socket is in connected state. There are some tasks for which this bit is never set. This flag is useful mostly to developers.</td>
</tr>
</tbody>
</table>
### General Concepts

#### Address and Prefix Formats

Advanced Routing Suite allows configuration of IPv4 address types only. Normally Advanced Routing Suite can recognize which type of address is being configured in a particular instance by the format of the address.

IPv4 addresses are 32 bits long. The formats of IPv4 addresses recognized by Advanced Routing Suite are:

- \( d \)
- \( d.d \)
- \( d.d.d \)
- \( d.d.d.d \)

where \( d \) represents a number in the range 0-255 inclusive. Each \( d \) specifies 8 bits of the address. If fewer than four \( d \) values are provided then the values provided specify the high order values of the address. For example, 192.168 is equivalent to 192.168.0.0.

In many cases IPv4 addresses are combined with masks to configure prefixes. There are two methods for specifying the mask: It can be specified as an IPv4 address proceeded by the `mask` keyword; or it can be specified as a length proceeded by the `masklen` keyword or, more conventionally, by a `/`. In the "`mask`" case, the address type of the mask must match the address type. Currently only contiguous bit masks are allowed in Advanced Routing Suite. Any non-zero address bits in positions that are covered by the specified mask cause a parse error. Example prefix specifications are:

- 10/8
- 10.0.0.0 mask 255.0.0.0 (equivalent to 10/8)
- 10 masklen 8 (equivalent to 10/8)
- 0/0 (IPv4 default address)
- 192.168.1/16 (invalid because the .1 is not covered by the mask)

#### Preferences Overview

Preference is the value that Advanced Routing Suite uses to select one route over another when more than one route to the same destination is learned from different protocols or peers. Preference can be set in the Advanced Routing Suite configuration files in several different configuration statements. Preference can be set based on one network interface over another, one protocol over another, or one remote gateway over another. Preference cannot be used to control the selection of routes within an interior gateway protocol. This control is accomplished automatically by the protocol based on metric. Preference can be used to select routes from the same exterior gateway protocol (such as BGP) learned from different peers or autonomous systems. Each route has only one configurable preference value associated with it, even though preference can be set at many places in the configuration file. Simply, the last or most specific
preference value set for a route is the value used. Preference can also be used to select one IGP instance over another.

The `preference` value is an arbitrarily assigned value used to determine the order of routes to the same destination in a single routing database. The active route is chosen by the lowest `preference` value. Some protocols implement a second preference (`preference2`), sometimes referred to as a tie-breaker. BGP and OSPF protocols use `preference2`. For OSPF, `preference2` is for internal use only and is not configurable. For BGP, `preference2` can be configured. Its value is used only when comparing routes with equal values of preference.

**Assigning Preferences**

A default preference is assigned to each source from which Advanced Routing Suite receives routes. Preference values range from 1 to 255, with the lowest number indicating the most preferred route.

*Note* - The default preference for direct routes (i.e., routes to subnets on directly connected interfaces) is 0. Other sources from which Advanced Routing Suite receives routes (i.e., OSPF) cannot be set to 0. The lowest preference value that can be specified for these is 1.

The following table summarizes the default preference values for routes learned in various ways. The table lists the statements (some of which are clauses within statements) that set preference and shows the types of routes to which each statement applies. The table lists the preference precedence between protocols and the default preference for each type of route. The more narrow the scope of the statement, the higher the precedence its preference value is given, but the smaller the set of routes it affects.

<table>
<thead>
<tr>
<th>Preference of</th>
<th>Defined by Statement</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directly connected networks</td>
<td><code>interface</code></td>
<td>0</td>
</tr>
<tr>
<td>Routes to interface aliases</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>OSPF routes</td>
<td><code>ospf</code></td>
<td>10</td>
</tr>
<tr>
<td>Redirects</td>
<td><code>redirect</code></td>
<td>30</td>
</tr>
<tr>
<td>Routes learned via route socket</td>
<td><code>kernel</code></td>
<td>40</td>
</tr>
<tr>
<td>Routes installed via SNMP</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>Routes learned via router discovery</td>
<td><code>router-discovery</code></td>
<td>55</td>
</tr>
<tr>
<td>RIP routes</td>
<td><code>rip</code></td>
<td>100</td>
</tr>
<tr>
<td>Point-to-point interface</td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>Routes to interfaces that are down</td>
<td><code>interface</code></td>
<td>120</td>
</tr>
<tr>
<td>Aggregate/generate routes</td>
<td><code>aggregate/generate</code></td>
<td>130</td>
</tr>
<tr>
<td>OSPF AS external routes</td>
<td><code>ospf/ospf3</code></td>
<td>150</td>
</tr>
<tr>
<td>BGP routes</td>
<td><code>bgp</code></td>
<td>170</td>
</tr>
<tr>
<td>Routes in kernel at startup</td>
<td></td>
<td>254</td>
</tr>
</tbody>
</table>
Chapter 3

Interfaces

In This Chapter

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interfaces Overview</td>
<td>42</td>
</tr>
<tr>
<td>autonomous-system</td>
<td>42</td>
</tr>
<tr>
<td>disable</td>
<td>43</td>
</tr>
<tr>
<td>preference</td>
<td>43</td>
</tr>
<tr>
<td>primary-alias</td>
<td>44</td>
</tr>
<tr>
<td>unnumbered</td>
<td>45</td>
</tr>
</tbody>
</table>

Interfaces Overview

Use interface commands to configure globally scoped interface parameters. With these commands, you can override system defaults for sets of interfaces. Both physical and logical interfaces can be individually identified and their global parameters set. In addition, sets of physical interfaces can be specified by way of a "wildcarding" mechanism.

autonomous-system

**Name**

`autonomous-system` - specifies the autonomous system (AS) used to create an AS path associated with the interface route created from this interface

**Syntax**

```
autonomous-system as-number
no autonomous-system as-number?
```

**Mode**

Interface Configuration

**Parameters**

*as-number* - an integer between 1 and 65535, inclusive

**Description**

The `autonomous-system` command specifies the autonomous system that will be used to create an AS path associated with the route created from the definition of this interface. The autonomous system number of the router running Advanced Routing Suite is specified with the global AS command. The autonomous system numbers of BGP's peers is specified in BGP configuration. An interface defines a route known as an "interface route". Interface routes are also known as direct routes. Specifying an AS number on the interface will cause the direct route corresponding to the interface to be generated with a non-empty AS path. If this route is then exported into BGP, update messages advertising this route will include the specified AS in their AS path. Normally, the interface AS number is not set.

The negative of this command, `no autonomous-system`, removes the autonomous-system definition previously defined by the autonomous-system command and reverts to the default of no autonomous-
system associated with this interface. Note: Specifying a value for as-number in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
By default, no AS number is associated with an interface.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example sets the AS for interface fxp1 to be 1439.

```
(config)# interface fxp1
(config-if)# autonomous-system 1439
```

disable

Name
disable – disables the associated interface

Syntax
disable
no disable

Mode
Interface Configuration

Parameters
none

Description
The disable command specifies that messages received on this interface must be ignored. The negative of this command, no disable, re-enables the interface.

Default
This option is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example disables interface 192.168.15.1

```
(config)# interface 192.168.15.1
(config-if)# disable
```

preference

Name
preference – used to select the best route when multiple routes exist for the same destination
Syntax

    preference pref
    no preference pref?

Mode
Interface Configuration

Parameters

`pref` - an integer in the range of 0 to 255 inclusive, with 0 being the highest preference that a route may have. **Note:** Only direct routes (in other words, routes to subnets on directly connected interfaces) can have a preference of 0.

Description

Multiple routes can exist for the same destination prefix. In this situation, the route’s preference is used to select the best route. For interface routes, `preference` sets the preference for routes to this interface when it is up.

The negative of this command, `no preference`, removes the configured preference and reverts to the default preference of 0. **Note:** Specifying a value for `pref` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `preference` is not configured, it is as if the user had configured the following:

```
(config-if)# preference 0
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example sets the preference for interface fxp1 to be 15.

```
(config)# interface fxp1
(config-if)# preference 15
```

primary-alias

Name

`primary-alias` - sets the primary logical address for a physical interface

Syntax

```
primary-alias [ ipv4-addr ipv4-address masklen len ]
no primary-alias [ ipv4-addr ipv4-address masklen len ]
```

Mode

Interface Configuration

Parameters

`ipv4-addr ipv4-address masklen len` - specifies an IPv4 address with a netmask. The value of `len` can be a number from 0 to 32, inclusive
Description

The `primary-alias` command is used to override the default selection of the primary logical interfaces on a given physical interface. There is exactly one primary interface for each subnet defined on a given physical interface. By default, Advanced Routing Suite selects the first logical read from the operating system for each subnet defined on the physical interface as a primary interface. The `primary-alias` command provides a way to specify which of the logical interfaces that share the same subnet is to be selected as primary for that subnet.

The negative of this command, `no primary-alias` removes the configured primary-alias command and reverts to the default primary addresses, which are the addresses of the first interfaces read from the kernel for each subnet on a physical interface. **Note:** Specifying parameters after `no primary-alias` has no effect on the configuration. Thus, they are displayed above as optional.

Default

The default primary addresses are the addresses of the first interfaces read from the kernel for each subnet on a physical interface.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures physical interface fxp1 and sets the `primary-alias` for that physical interface to 192.168.10.15 with a mask length of 24.

```
(config)# interface fxp1
(config-if)# primary-alias ipv4-addr 192.168.10.15 masklen 24
```

unnumbered

**Name**

`unnumbered` - configures the specified interface as an unnumbered interface

**Syntax**

```
unnumbered interface-name

no unnumbered interface-name
```

**Mode**

Interface Configuration

**Parameters**

`interface-name` - the name of the physical interface from which an interface address will be borrowed

**Description**

The `unnumbered` command indicates that the associated interface is an "unnumbered" interface. The purpose of unnumbered interfaces is to allow a network administrator to save IP address space. There are, by definition, only two machines on a serial or ppp link, and having to assign an IP subnet is an inefficient use of IP address space. In order to forward packets, a host or router must have an IP source address.

The IPv4 address of the first primary logical interface on the physical interface named by the `interface-name` parameter will be borrowed and used for the interface named in this command.

The term, "Unnumbered Interface" is actually a misnomer. The interfaces are not really unnumbered. What actually happens is the unnumbered interface borrows an IP address from another configured interface on the box to use as its source address.

With regards to unnumbered interfaces, the following assumptions are made:
- The p2p or serial link has a framing protocol for encapsulating IP packets (such as PPP or HDLC).
- The operating system provides a method for joining a multicast group on a physical interface by referring to the interface by name (such as "ppp1") or interface index.
- A supporting physical interface exists on the box that is configured with a valid IP address.
- A route can be installed in the OS forwarding table with a nexthop address of 0 or a nexthop specifying the physical interface.

The negative of this command, no unnumbered, removes the unnumbered definition previously configured with the unnumbered command and reverts to the default of the interface being a numbered interfaces.

**Default**

Interfaces are not unnumbered by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures interface eth0 as an unnumbered interface.

```plaintext
(config)# interface eth0
(config-if)# unnumbered eth1
```
Chapter 4

Kernel Interface

In This Chapter

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel Interface Overview</td>
<td>47</td>
</tr>
<tr>
<td>kernel background limit</td>
<td>47</td>
</tr>
<tr>
<td>kernel background priority</td>
<td>48</td>
</tr>
<tr>
<td>kernel flash limit</td>
<td>49</td>
</tr>
<tr>
<td>kernel flash type</td>
<td>50</td>
</tr>
<tr>
<td>kernel no-change</td>
<td>51</td>
</tr>
<tr>
<td>kernel no-flush-at-exit</td>
<td>51</td>
</tr>
<tr>
<td>kernel no-install</td>
<td>52</td>
</tr>
<tr>
<td>kernel remnant-holdtime</td>
<td>53</td>
</tr>
<tr>
<td>kernel routes</td>
<td>54</td>
</tr>
<tr>
<td>kernel trace file</td>
<td>55</td>
</tr>
<tr>
<td>kernel trace flag</td>
<td>56</td>
</tr>
<tr>
<td>show kernel</td>
<td>57</td>
</tr>
</tbody>
</table>

Kernel Interface Overview

Although the kernel interface is not technically a routing protocol, it has many characteristics of one, and Advanced Routing Suite handles it similarly. The routes Advanced Routing Suite chooses to install in the kernel forwarding table are those that will actually be used by the kernel to forward packets.

The add, delete, and change operations that Advanced Routing Suite must use to update the typical kernel forwarding table take a non-trivial amount of time. The time used does not present a problem for older routing protocols (such as RIP), which are not particularly time critical and do not easily handle large numbers of routes anyway. The newer routing protocols (such as OSPF and BGP) have stricter timing requirements and are often used to process many more routes. The speed of the kernel interface becomes critical when these protocols are used.

To prevent Advanced Routing Suite from locking up for significant periods of time while installing large numbers of routes (up to a minute or more has been observed on real networks), the processing of these routes is done in batches. The size of these batches can be controlled by the tuning parameters shown below, but normally the default parameters will provide the proper functionality.

During normal shutdown processing, Advanced Routing Suite deletes all the routes it has installed in the kernel forwarding table, except for those static routes marked with flag-retain. (See Chapter 8 (Static Routes) for more information.) Optionally, Advanced Routing Suite can leave all routes in the kernel forwarding table by not deleting any routes using no-flush-at-exit. This option is useful on systems with large numbers of routes because it eliminates the need to re-install the routes when Advanced Routing Suite restarts, which can greatly reduce the time it takes to recover from a restart.

kernel background limit

Name

kernel background limit - specifies the type of routes that will be processed during a flash update
Kernel Interface

Syntax

```
kernel background limit number
no kernel background limit number?
```

Mode

Global Configuration

Parameters

`number` - specifies a number of routes that can be processed during one batch. The valid value range is 0 to 4,294,267,295, inclusive.

Description

Because only interface routes are normally installed during a flash update, the remaining routes are processed in batches in the background, that is, when no routing protocol traffic is being received. The `kernel background limit` command specifies how these batches are processed.

The negative of this command, `no kernel background limit`, reverts back to the default of maximum of 120 routes of lower priority. **Note:** Specifying a value for `number` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `kernel background` is not specified, it is the same as if the user had specified the following:

```
(config)# kernel background limit 120
```

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, the number of background route updates is limited to 40 routes.

```
(config)# kernel background limit 40
```

kernel background priority

Name

`kernel background priority` - specifies the priority of the processing of batches of operating system updates in relationship to the flash update processing

Syntax

```
kernell background priority [lower | higher | flash]
no kernel background priority [lower | higher | flash]?
```

Mode

Global Configuration

Parameters

`lower` - flash updates are processed first
`higher` - operating system updates are processed first
`flash` - operating system updates are processed that the same priority as flash updates
Description
Because only interface routes are normally installed during a flash update, the remaining routes are processed in batches in the background, when no routing protocol traffic is being received. The kernel background priority command along with the kernel background limit command specifies how these batches are processed.

The kernel background priority command specifies the priority of the processing of batches of operating system updates in relationship to the flash update processing. The negative form of this command, no kernel background priority, removes the configured priority value, and returns this to its default value of lower. **Note:** Specifying a priority value in the no form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default
If kernel background priority is not specified, it is the same as if the user had specified the following:

```
(config)# kernel background priority lower
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example sets the background priority to flash, thereby giving operational system updates the same priority as flash updates.

```
(config)# kernel background priority flash
```

**kernel flash limit**

**Name**
kernel flash limit — specifies the number of routes processed during a flash update

**Syntax**

```
kernelflashlimit [number | none]
```

**Mode**
Global Configuration

**Parameters**

*number | none* — specifies the maximum number of routes that can be processed during a flash update or specify that an unlimited number are processed. The valid value range for number is from 0 to 4,294,967,294, inclusive.

**Description**
A flash update results from protocol activity. The kernel flash limit command controls the number of routes installed during a flash update, suspending the current protocol module until the flash completes. Typically, only up to twenty interface routes are normally installed during a flash update. The remaining routes are processed in batches in the background, when no routing protocol traffic is being processed.

The negative of this command, no kernel flash, reverts back to the default of maximum of 20 routes of type interface. **Note:** Specifying a limit value in the no form has no effect on the configuration. Thus, it is displayed above as optional.
Default
If `kernel flash limit` is not specified, it is the same as if the user had specified the following:

```
(config)# kernel flash limit 20
```

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, the router is configured to flash up to 40 routes at a time.

```
(config)# kernel flash limit 40
```

**kernel flash type**

Name
`kernel flash type` - specifies the type of routes that will be processed during a flash update

Syntax
```
kernl flash type [ interface | interior | all ]
no kernel flash type [ interface | interior | all ]?
```

Mode
Global Configuration

Parameters
- `interface` - specifies routes defined by an interface
- `interior` - specifies routes within the domain
- `all` - specifies all routes

Description
When routes change as a result of operating system or protocol module activity, the process of notifying the Advanced Routing Suite protocol module is known as a "flash update." The `kernel flash type` command specifies the type of routes that will be processed during a flash update. The operating system’s forwarding table interface is the first to be notified. The flash process is concerned with the following three types of routes:

- interface routes - routes defined by an interface
- interior routes - routes within the domain
- all routes - routes defined within the domain and routes defined by an interface

The negative form of this command, `no kernel flash type`, resets the flash type to `interface`. **Note:** Specifying a type value in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `kernel flash type` is not specified, it is the same as if the user had specified the following:

```
(config)# kernel flash type interface
```

Command History
NGC 2.2 - This command was introduced.
Examples
The following example specifies to process all routes during a flash update.

(config)# kernel flash type all

kernel no-change

Name
eKernel no-change - determines whether change operations will be performed

Syntax

   kernel no-change
   no kernel no-change

Mode
Global Configuration

Parameters
none

Description
On systems supporting the routing socket, the kernel no-change command determines whether change operations will be performed. If the command is configured, then change operations will not be performed, only deletes and adds will. The negative of this command, no kernel no-change, reverts back to the default of change operations being performed.

Default
By default, kernel no-change is disabled.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns kernel no-change on.

(config)# kernel no-change

kernel no-flush-at-exit

Name
Kernel no-flush-at-exit - determines whether deletions occur during shutdown

Syntax

   kernel no-flush-at-exit
   no kernel no-flush-at-exit

Mode
Global Configuration
Parameters
none

Description
During a Advanced Routing Suite shutdown/restart sequence, it may be desirable to keep the routes that existed at the time of the Advanced Routing Suite shutdown in the operating system’s forwarding table. This enables the router to continue forwarding packets while Advanced Routing Suite is being restarted. After a restart, the protocol modules are given a short amount of time (currently three minutes) to determine their routes. After three minutes, all residual routes not re-established by the protocol modules are flushed.

The following are four conditions under which Advanced Routing Suite does not flush a route:

- The route is an interface route.
- The route is a static route configured within a static retain command.
- The kernel no-flush-at-exit command is configured.
- The route’s static bit is set.

The kernel no-flush-at-exit command is handy on systems with thousands of routes. Upon startup, Advanced Routing Suite will notice which routes are in the operating system’s forwarding table and not add them back.

The negative of this command, no kernel no-flush-at-exit, reverts back to the default of disabled.

Default
By default, kernel no-flush-at-exit is disabled.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns kernel no-flush-at-exit on.

```
(config)# kernel no-flush-at-exit
```

See Also
Chapter 8 Static Routes

kernel no-install

Name
kernel no-install - specifies to no install routes in the kernel Forwarding Information Base (FIB)

Syntax
```
kernel no-install
no kernel no-install
```

Mode
Global Configuration

Parameters
none
**Description**

The `kernel no-install` command specifies to not install routes in the kernel FIB. Normally, the route with the lowest preference is installed in the kernel forwarding table and is the route exported to other protocols. When this command is configured, the associated static route will not be installed in the forwarding table when it is active, but it will still be eligible to be exported to other protocols.

The negative of this command, `no kernel no-install`, reverts back to the default of disabled.

**Default**

By default, `kernel no-install` is disabled.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example turns `kernel no-install` on.

```
(config)# kernel no-install
```

---

**kernel remnant-holdtime**

**Name**

`kernel remnant-holdtime` - sets the holdtime for remnant routes

**Syntax**

```
kernell remnant-holdtime time-seconds
```

**Mode**

Global Configuration

**Parameters**

`time-seconds` - an integer between 0 and 900, inclusive

**Description**

When Advanced Routing Suite starts up, it reads the kernel forwarding table and installs corresponding routes into Advanced Routing Suite’s routing table. These routes, with the exclusion of interface routes and routes configured via the UNIX route command, are called "remnants." Remnant routes are timed out after the specified interval, or as soon as a more attractive route is learned. This method allows forwarding to occur while the routing protocols start learning routes.

The negative of this command, `no kernel remnant-holdtime`, reverts back to the default of 180 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `kernel-remnant holdtime` is not specified, it is the same as if the user had specified the following:

```
(config)# kernel remnant-holdtime 180
```

**Command History**

NGC 2.2 - This command was introduced.
Examples
The following example changes the remnant-holdtime to 90 seconds.

(config)# kernel remnant-holdtime 90

kernel routes

Name
kernel routes - limits the maximum number of routes Advanced Routing Suite will install in the kernel

Syntax

    kernel routes limit
    no kernel routes limit?

Mode
Global Configuration

Parameters

limit - an integer in the range of 0 to 4,294,967,295, inclusive, on 32-bit systems and 0 to 264-1, inclusive, on 64-bit systems

Description
On some systems, kernel memory is at a premium. With the kernel routes command, a limit can be placed on the maximum number of routes Advanced Routing Suite will install in the kernel. This discussion is concerned with three types of routes:
- interface routes - routes defined by an interface configuration (includes UNIX ‘ifconfig’ and ‘route’ generated routes)
- interior routes - routes within the domain
- exterior routes - routes exterior to the domain

Normally, Advanced Routing Suite adds, changes, or deletes routes in interface/interior/exterior order. That is, Advanced Routing Suite queues interface routes first, followed by interior routes, followed by exterior routes, and then processes the queue from the beginning. When the route limit is reached, Advanced Routing Suite must ensure that interface/interior/exterior route preferences are followed. This is accomplished by first deleting kernel-based routes and then turning queued changes into adds. Finally, the list of active routes in the RIB is processed in interface/internal/external order, until the route limit is reached.

The negative of this command, no kernel routes, reverts back to the default of 232-1 (4,294,967,295) routes for 32-bit systems and 264-1 routes for 64-bit systems. Note: Specifying a value for num in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
For 32-bit systems, if kernel routes is not specified, it is the same as if the user had specified the following:

(config)# kernel routes 4294967295

For 64-bit systems, if kernel routes is not specified, it is the same as if the user had specified the following:

(config)# kernel routes 18446744073709551616

Command History
NGC 2.2 - This command was introduced.
Examples
The following example limits the number of routes to 500.

```
(config)# kernel routes 500
```

**kernel trace file**

**Name**
kernel trace file - specifies file options when tracing in the kernel interface

**Syntax**
```
kernl trace file file_name [no-timestamp || overwrite]?
```

**Mode**
Global Configuration

**Parameters**
file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
no-timestamp - specifies that a timestamp should not be prepended to all trace lines
overwrite - specifies to begin tracing by appending or truncating an existing file

**Description**
The trace file command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. The kernel trace file command specifies a file for tracing of all Kernel events. The negative form of this command disables this tracing. The specific events that are traced are controlled. The no-timestamp option disables the prepending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The overwrite option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands. Consider the following example:

```
(config)# kernel trace file /var/log/kernel.log
no-timestamp
(config)# kernel trace file /var/log/kernel.log
max-files 10
```

The option given in the second command completely replaces that given in the first. In order to specify both no-timestamp and max-files 10, they must be entered on the same line as follows.

```
(config)# kernel trace file /var/log/kernel.log
max-files 10 no-timestamp
```

**Default**
Kernel tracing is turned off by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
In the following example, kernel tracing is written to the file "/var/tmp/kernel.log". No timestamp will display at the beginning of the trace lines.

```
(config)# kernel trace file /var/tmp/kernel.log no-timestamp
```
kernel trace flag

Name
kernel trace flag - specifies Kernel-specific tracing options as well as options that are common across all protocols

Syntax

```
kernel trace flag ([ route | normal | state | policy | task | timer | all ] ) | ([ symbol | remnants | interface-list | request | info ] ) |
([ packets | redirect | other | routes | interface ] [ send | receive | send-receive ]? [detail?] )
```

```
no kernel trace flag ([ route | normal | state | policy | task | timer | all ] ) | ([ symbol | remnants | interface-list | request | info ] ) |
([ packets | redirect | other | routes | interface ] [ send | receive | send-receive ]? [detail?] )
```

Mode
Global Configuration

Parameters
Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. Note: Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

Kernel-specific flags that do not allow send, receive, send-receive, or detail actions:

```
[ symbol | remnants | interface-list | request | info ] - These Kernel-specific flags cannot be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```

- **symbol** - specifies to trace symbols, which are read from the kernel. The only useful way to specify this level of tracing is via the Advanced Routing Suite `-t` option, because the symbols are read from the kernel before parsing the configuration file.
- **remnants** - specifies to trace remnants, or routes read from the kernel when Advanced Routing Suite starts
- **interface-list** - specifies to trace the interface list. `interface-list` is useful when specified with the Advanced Routing Suite `-t` option because the first interface list scan is performed before any configuration information is processed.
- **request** - specifies to trace requests that specify to add, delete, or change routes in the kernel forwarding table
• **info** - specifies to trace info messages, which are messages received from the routing socket, such as TCP lossage, routing lookup failure, and route resolution requests. Advanced Routing Suite does not currently process these messages, but logs the information if requested.

Kernel-specific flags that allow associated actions:

```
[packets | redirect | other | routes | interface] - These Kernel-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```

- **packets** - specifies to trace all kernel packet types
- **redirect** - specifies to trace redirect messages, which are received from the kernel
- **other** - specifies to trace other messages that are received from the kernel, including those mentioned in the info type above. This option is currently not being used and is reserved for future use.
- **routes** - specifies to trace routes that are exchanged with the kernel, including add, delete, or change messages and add, delete, or change messages received from other processes
- **interface** - specifies to trace interface status messages that are received from the kernel. These are supported only on systems with networking code derived from BSD 4.4.

```
[send | receive | send-receive]? - optionally specify whether to limit the tracing to packets sent, received, or both
```

```
[detail?] - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines
```

**Description**

Use the `kernel trace flag` command to specify tracing flags for Kernel tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both symbol and remnant packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, trace flags specify that both the sent and received redirect and interface messages are traced in detail. This tracing information will be written to the file `/var/tmp/kernel.log`.

```
(config)# kernel trace file /var/tmp/kernel.log
(config)# kernel trace flag redirect send-receive detail
(config)# kernel trace flag interface send-receive detail
```

**show kernel**

**Name**

`show kernel` - displays information about kernel settings and support

**Syntax**

```
show kernel
```

**Mode**

User Execution
Parameters

none

Description

The show kernel query displays information about kernel settings and support. These fields have the following meanings.

no-change - If this setting is set to "on" for systems supporting the routing socket, then change operations will not be performed. Only deletes and adds will be performed. This is useful on early versions of the routing socket code where the change operation was broken. This option can be toggled with the kernel no-change command.

no-flush-at-exit - During normal shutdown processing, Advanced Routing Suite deletes from the kernel forwarding table all routes that the kernel does not flag as "retain". Setting this option to "on" prevents route deletions at shutdown. During a Advanced Routing Suite shutdown/restart sequence, it may be desirable to keep in the kernel forwarding table the routes that existed at the time of the Advanced Routing Suite shutdown. This allows the router to continue forwarding packets while Advanced Routing Suite is being restarted.

After a restart, the protocol modules are given a short amount of time (currently three minutes) to determine their routes. After three minutes, all residual routes not re-established by the protocol modules are flushed.

There are four conditions under which Advanced Routing Suite does not flush a route:

- Interface routes
- Static routes with "retain" specified
- Routes specified with "noflushexit"
- Routes with static bit set

The no-flush-at-exit option is convenient on systems with thousands of routes. Upon startup, Advanced Routing Suite will notice the routes that are in the kernel's forwarding table and not add them back.

This option can be toggled with the kernel no-flush-at-exit command.

reject support - This value indicates whether the operating system supports reject routes.

blackhole support - This value indicates whether the operating system supports blackhole routes.

variable length subnet support - This value indicates whether the operating system supports data forwarding using variable length subnet masks.

host route support - This value indicates whether the operating system supports hosts routes.

multipath support - This value indicates whether the operating system supports multipath routing.

IPv4 forwarding enabled - This value indicates whether IPv4 forwarding is turned on in the operating system.

UDP checksums enabled - This value indicates whether UDP checksums are turned on in the operating system.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example displays a response for the show kernel query.

> show kernel
no-change: on           no-flush-at-exit: off
reject support: yes    blackhole support: yes
variable length subnet support: no
host route support: yes
multipath support: yes
IPv6 forwarding enabled: no
IPv4 forwarding enabled: yes
UDP checksums enabled: no
Chapter 5

Martian Addresses

In This Chapter

Martian Addresses Overview 60
martian 60

Martian Addresses Overview

Martians are networks that are considered illegal to be routed on the Internet. The `martian` command allows additions to the list of martian addresses. An `accept` option can also be specified to explicitly allow a subset of a range that was disallowed. The `martian` command can also be used for route filtering. Any prefixes that are declared to be martians will be automatically filtered out from all received and advertised routes. However, routes already in the routing table will not be removed if declared as martian until the next neighbor or protocol reset (for example, a BGP neighbor soft reconfig).

RFC 1918 specifies these networks as part of the private Internet space:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

The prefixes are considered un-routable between autonomous systems. However, these prefixes can be routed within autonomous systems. Advanced Routing Suite does not treat these as martian addresses, but the `martian` command will allow you to treat private address space as illegal for routing within an autonomous system. RFC 1700 specifies common usage for IP numbers.

The default list of martians is:

- 127/8 (127.0.0.0 netmask 255.0.0.0) - 127.x.x.x is specified by RFC 1700 to loop back addresses. RFC 1700 (page 4, item g) states “these addresses should never appear outside a host”. Address 127.0.0.1 is normally used as a loopback address.
- 224/3 - these are the multicast addresses.

martian

Name

`martian` - configures a martian address

Syntax

```
martian ipv4-addr ipv4-address [ mask mask | masklen masklen ] [ accept | reject ] [ (ge length) || (le length) ]?
```

```
no martian ipv4-addr ipv4-address [ mask mask | masklen masklen ] [ accept | reject ] [ (ge length) || (le length) ]?
```

Mode

Global Configuration
Parameters

IPv4

ipv4-addr ipv4-address - specify ipv4-addr with an address in dotted-quad format that, when combined with mask or masklen, specifies a set of martian addresses

mask mask | masklen masklen - specify either an integer mask length (from 0 to 32, inclusive) or specify a contiguous mask in dotted-quad format

accept | reject - specify whether the prefix should be allowed to be routed or should be rejected

ge length | reject - optionally specify that the configured martian matches prefix and mask pairs with exactly the same prefix address and with mask lengths that are greater than or equal to this value. The value range for length is 0 to 32, inclusive. Additionally, it must be at least the value of masklen (or the number of significant bits in mask) and no greater than the value of le.

le length - optionally specify that the configured martian matches prefix and mask pairs with exactly the same prefix address and with mask lengths that are less than or equal to this value. The value range for length is 0 to 32, inclusive. Additionally, it must be at least the value of masklen (or the number of significant bits in mask) and no less than the value of ge.

Description

Use the martian command to configure one or more martian addresses. Martians are networks that are considered illegal to be routed on the Internet. This command allows you to specify private Internet space as un-routable. Conversely, you can use the accept option to explicitly allow a subset of a range that was disallowed.

If neither ge nor le is specified, then ge and le default to the configured masklen (or to the IPv4 contiguous mask value).

If ge is specified, but le is not, then le defaults to the maximum mask length for the address family.

If le is specified but ge is not, then ge defaults to masklen.

Note: Order is not important when specifying ranges. For example, you may want to configure a range to be unroutable, then configure a more specific range within that range to be routable. The lookup always finds the most specific entry (the match with the longest prefix) regardless of the order in which the entries were entered.

Default

The default list of martians include:

- 127/8 - specified by RFC 1700 to loop back addresses. RFC 1700 (page 4, item g) states "these addresses should never appear outside a host." Address 127.0.0.1 is normally used as a loopback address.
- 224/3 - the multicast addresses

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example configures 35/8 as a martian prefix.

```
(config)# martian ipv4-addr 35.0.0.0 masklen 8 reject
```

Example 2

In this example, the IPv4 address 127.168.14.15, which would normally be considered a martian route per RFC 1918, is removed from the martian list.
(config)# martian ipv4-addr 127.168.14.15 masklen 32 accept

Example 3
The following example configures the IPv4 addresses within 32.0.0.0, with a mask-length of 8 and a prefix length of at least 16, as martian addresses.

(config)# martian ipv4-addr 32.0.0.0 masklen 8 reject ge 16
Chapter 6

Multicast

In This Chapter

Multicast Overview 63
clear ip mroute 63
ip multicast boundary 64
ip multicast ttl-threshold 64
show ip mroute 65
show ip multicast boundary 66
show ip multicast ttl-threshold 67

Multicast Overview

The multicast commands are used to set interface-specific options such as time-to-live (TTL) thresholds and administratively scoped boundaries.

clear ip mroute

Name

clear ip mroute - clears routes in the multicast routing table

Syntax

clear ip mroute gr_address? sr_address?

Mode

User Execution

Parameters

gr_address - optionally specify a valid IPv4 multicast group address that you want removed from the multicast routing table
sr_address - optionally specify a valid IPv4 multicast source address that you want removed from the multicast routing table

Description

Use the clear ip mroute command to remove routes from the multicast routing table. If this command is issued without arguments, then all group and source information will be cleared. Similarly, you can specify to clear a specific group and/or source multicast address.

Command History

NGC 2.2 - This command was introduced.
Examples
The following example clears group/source 226.1.1.1/192.168.0.1 from the multicast routing table.

> clear ip mroute 226.1.1.1 192.168.0.1

ip multicast boundary

Name
ip multicast boundary - specifies an administratively scoped boundary for a single multicast group on the associated interfaces.

Syntax

ip multicast boundary group group-address masklen length
no ip multicast boundary group group-address masklen length

Mode
Interface Configuration

Parameters

- group group-address - specify a valid IPv4 address to denote the group address
- masklen length - the length of the mask associated with the group prefix, specified as an integer

Description
ip multicast boundary is used to configure administratively scoped group boundaries on the indicated interface(s).

Default
The ip multicast boundary command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, an administratively scoped boundary for multicast group 224.5.5.5 is configured.

(config-if)# ip multicast boundary group 224.5.5.5
masklen 24

ip multicast ttl-threshold

Name
ip multicast ttl-threshold - specifies the minimum time-to-live that a multicast data packet can have and still be forwarded over the associated interface

Syntax

ip multicast ttl-threshold ttl
no ip multicast ttl-threshold [ ttl ]?

Mode
Interface Configuration
Parameters

ttl - an integer from 0 to 255, inclusive

Description

The ip multicast ttl-threshold command specifies the minimum time-to-live plus 1 that a multicast data packet can have and still be forwarded over the associated interface. A value of 0 indicates that no multicast data packets should be forwarded over the associated interface(s).

Default

If ip multicast ttl-threshold is not configured, it is the same as if the user had specified the following:

(config-if)# ip multicast ttl-threshold 0

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a threshold of 20 on interface fxp1.

(config)# interface fxp1
(config-if)# ip multicast ttl-threshold 20

show ip mroute

Name

show ip mroute - displays the contents of the Multicast Routing Table

Syntax

show ip mroute

Mode

User Execution

Parameters

none

Description

The output of the show ip mroute query displays the content of the Multicast routing table.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example shows a response to the show ip mroute query.

> show ip mroute
IP Multicasting Routing Table
(*, 224.1.1.1), uptime 0:01:20
Incoming interface: fxp0, RPF neighbor 10.2.11.31
Outgoing interface list:
fxp1
Field Descriptions
The following table describes the fields that appear in the Multicast MRT Query.

Table 11-1  Multicast MRT Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(*, 224.1.1.1), (192.168.101.100, 224.2.2.2)</td>
<td>The entries in the IP multicast routing table.</td>
</tr>
<tr>
<td>Uptime</td>
<td>The length of time that the (*,G) or (S,G) entry has been created in hours:minutes:seconds.</td>
</tr>
<tr>
<td>Incoming interface</td>
<td>The expected interface for a multicast packet from the source.</td>
</tr>
<tr>
<td>RPF neighbor</td>
<td>The IP address of the upstream router to the source.</td>
</tr>
<tr>
<td>Outgoing interface list</td>
<td>The interfaces through which packets will be forwarded.</td>
</tr>
</tbody>
</table>

show ip multicast boundary

Name
show ip multicast boundary  - displays all boundaries within all interfaces

Syntax

    show ip multicast boundary

Mode
User Execution

Parameters
none

Description
Use the show ip multicast boundary query to obtain summarized information for all boundaries within all interfaces.

Command History
NGC 2.2 - This command was introduced.
Examples
The following example shows a response to the `show ip multicast boundary query`.

```
> show ip multicast boundary
[1]fxp0, 224.5.5.0/24
[2]fxp0, 224.5.10.0/24
[3]fxp1, 239.5.0.0/16
```

**show ip multicast ttl-threshold**

**Name**
`show ip multicast ttl-threshold` - displays information about the multicast TTL threshold

**Syntax**
```
show ip multicast ttl-threshold
```

**Mode**
User Execution

**Parameters**
one

**Description**
The output of the `show ip multicast ttl-threshold` query displays information about the multicast TTL threshold.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example shows the output of a `show ip multicast ttl-threshold` query.

```
> show ip multicast ttl-threshold
  fxp0, 1
  fxp2, 5
```

**Field Descriptions**
The following table describes the fields that appear in the Multicast TTL Threshold Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fxp0, 1 and fxp1, 5</td>
<td>Shows the ttl-threshold value for each interface.</td>
</tr>
</tbody>
</table>
Chapter 7

Trace Options

In This Chapter

Trace Options Overview 68
trace file 68
trace flag 69

Trace Options Overview

The `trace` command controls Advanced Routing Suite’s tracing options. These options can be configured at many levels. Tracing options include the file specifications and global and protocol-specific tracing options. Unless overridden, tracing options from the next higher level are inherited by lower levels. For example, BGP peer tracing options are inherited from BGP group tracing options, which are inherited from global BGP tracing options, which are inherited from global Advanced Routing Suite tracing options. At each level, additional tracing specifications override the inherited options.

Global tracing configurations have an immediate effect. Tracing values inherited by protocols specified in a configuration are initially inherited from the global options that are currently in effect as the protocol configuration entries are entered, unless they are overridden by more specific options.

The `file` options in subsequent `trace` configurations with the same file name modify the previously set values for the files.

trace file

Name

`trace file` - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed

Syntax

```
trace file file_name [no-timestamp || overwrite]?
no trace file file_name [no-timestamp || overwrite]?
```

Mode

Global Configuration

Parameters

`file_name` - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.

`no-timestamp` - specifies that a timestamp should not be prepended to all trace lines

`overwrite` - specifies to begin tracing by appending or truncating an existing file
Description
This command specifies a file for tracing of all protocols. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `trace flag` command. The `trace file` command is also associated with each protocol, so that information pertaining to a single protocol can be written to its own file.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands.

Default
Global tracing options are not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, tracing is written to the file "gated.log". No timestamp will display at the beginning of the trace lines.

```
(config)# trace file gated.log no-timestamp
```

trace flag

**Name**
`trace flag` - specifies global tracing options that are common across all protocols

**Syntax**
```
trace flag ( [ route | normal | state | policy | task | timer | all ] )
no trace flag ( [ route | normal | state | policy | task | timer | all ] )
```

**Mode**
Global Configuration

**Parameters**

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```
- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags
Description
Use the `trace flag` command to specify global tracing flags. Each flag must reside on its own configuration line. For example, you cannot specify to trace both task and policy packets in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, flags specify to trace both timer usage and state machine transition. This tracing information will be written to the file gated.log.

```
(config)# trace file gated.log
(config)# trace flag timer
(config)# trace flag state
```
Chapter 8

Border Gateway Protocol (BGP)

In This Chapter

Border Gateway Protocol (BGP) Overview 73
address-family 77
bgp always-compare-med 78
bgp as-path-loops 78
bgp bestpath as-path ignore 79
bgp bestpath compare-cluster-list-length 80
bgp bestpath compare-originator-id 81
bgp bestpath compare-router-id 82
bgp bestpath med confed 82
bgp bestpath med missing-as-worst 83
bgp cluster-id 84
bgp confederation identifier 85
bgp confederation peers 86
bgp non-leading-confeds 86
bgp open-on-accept 88
bgp pass-optional-nontrans 88
bgp restart-defer 89
bgp restart-delete-remnants 90
bgp restart-time 91
bgp restart-timeout 91
bgp router-id 92
bgp send-group-always 93
bgp tie-break-on-age 94
clear ip bgp 94
default-metric 95
distance 96
distribute-list 97
enable 98
maximum-routes 99
neighbor add-communities 100
neighbor aggregator-id 101
neighbor allow 102
neighbor as-loop 103
neighbor as-override 104
neighbor aspath-prepend 105
neighbor capability orf comm-filter 105
neighbor capability orf extcomm-filter 106
neighbor capability orf prefix-filter 107
neighbor cluster-id 108
neighbor distance 109
neighbor dynamic 110
Border Gateway Protocol (BGP) Overview

The Border Gateway Protocol (BGP) is an exterior, or inter-domain, routing protocol. BGP is used to exchange routing information between multiple transit autonomous systems, between transit and stub autonomous systems, or between two stub autonomous systems.

BGP uses path attributes to provide more information about each route and in particular to maintain an autonomous system (AS) path. An AS path includes the AS number of each autonomous system that the route has transited, which provides information sufficient to prevent routing loops in an arbitrary topology. Path attributes can also be used to distinguish between groups of routes to determine administrative preferences, allowing greater flexibility in determining route preference to achieve a variety of administrative ends. Advanced Routing Suite supports version 4 of the BGP protocol.

BGP supports two basic types of sessions between neighbors: internal (sometimes referred to as IBGP) and external (EBGP). Internal sessions are run between routers in the same autonomous system. External sessions run between routers in different autonomous systems. When a router routes to an external peer, the local AS number is prepended to the AS path. This means that routes received from an external peer are guaranteed to have the AS number of that peer at the start of the path. In general, routes received from an internal neighbor will not have the local AS number prepended to the AS path. Those routes will have the same AS path that the route had when the first internal neighbor received the route from an external peer. Routes with no AS numbers in the path may be legitimately received from internal neighbors. These routes should be considered internal to the receiver’s own AS.

External BGP sessions may or may not include a single metric, which BGP calls the Multi-Exit Discriminator (MED) among the path attributes. MED is a 32-bit unsigned integer. Smaller values of the MED are preferred. This metric is used only to break ties between routes with equal preference from the same neighboring AS.

Internal BGP sessions carry at least one metric in the path attributes, which BGP calls the Local_Pref. A route is preferred if its value for this metric is larger. Internal sessions can optionally include a second metric, the MED, carried in from external sessions.

BGP collapses as many routes with similar path attributes as it can into a single update for advertisement. The churn caused by the loss of a neighbor will be minimized, and the initial advertisement sent during peer establishment will be maximally compressed. BGP does not read information from the kernel message by message, but fills the input buffer. BGP processes all complete messages in the buffer before reading again. BGP also does multiple reads to clear all incoming data queued on the socket. This feature may cause other protocols to be blocked for prolonged intervals by a busy peer connection. All unreachable messages are collected into a single message and sent prior to reachable routes during a flash update. Another update is sent once the maximum packet size is reached.

Advanced Routing Suite will determine the immediate next hop to use for a next hop sent in a BGP update by finding the best matching (most specific) route in the routing table that covers the address of the BGP next hop. This allows BGP to support distant BGP peers that might not be directly connected.

BGP allows unconfigured peers to connect if an appropriate group has been configured with a “neighbor allow” (“neighbor allow” on page 102) command.

Notes:
1. The peer-group command is used to create peer groups that share common attributes.
2. Within the peer-group command, the neighbor commands are used to individually specify peers and to permit overriding specific peering group options.
3. When comparing routes with the same MEDs, BGP first prefers routes learned from external routers (EBGP), followed by confederation external routers, over routes learned from internal routers (IBGP).

BGP Decision Process

BGP uses the following decision process when selecting between multiple routes to the same destination:
1. Prefer the route with the lowest Advanced Routing Suite preference.
2. Prefer the route with the lowest Advanced Routing Suite preference.
3. Prefer the route with the highest LOCAL_PREF value.
4. Prefer the route with the shortest path, excluding confederation segments.
5. Prefer the route with the “best” ORIGIN: IGP is better than EGP which is better than incomplete.
6. If "bgp always-compare-med" ("bgp always-compare-med" on page 78) has not been configured, prefer any routes that do not have an inferior MED. Routes are considered to have an inferior MED if two routes to the same destination have been learned with the same neighbor AS, and one of the routes being considered does not have the lowest MED received for routes to that destination from the same neighbor AS. If confed-MED is specified, neighbor AS is considered to be the first AS in the AS_PATH, or the local AS if there are no AS numbers in the AS_PATH. If confed-MED is not specified, neighbor AS is considered to be the first AS in the non-confed segment of the AS_PATH, or the local confederation-id if there are no AS numbers in a non-confed segment of the AS_PATH.

If "bgp always-compare-med" ("bgp always-compare-med" on page 78) has been configured, prefer the route with the lowest MED.
7. Prefer the route with the lowest IGP cost to the BGP nexthop. IGP cost is determined by comparing the preference, then the preference 2, then the metric, and finally the metric 2 of the two resolving routes.
8. Prefer routes received from external peers.
9. If "bgp tie-break-on-age" ("bgp tie-break-on-age" on page 94) has been specified, prefer the older route.
10. If "bgp bestpath compare-router-id" ("bgp bestpath compare-router-id" on page 82) has been specified, prefer the route learned with the lowest router ID. The router ID is taken from the Open message of the peering session over which the route was received, unless "bgp bestpath compare-originator-id" ("bgp bestpath compare-originator-id" on page 81) has been specified, and the route was received with an ORIGIN_ID. In the latter case, the ORIGIN_ID is used instead of the router ID from the Open message.
11. If "bgp bestpath compare-cluster-list-length" ("bgp bestpath compare-cluster-list-length" on page 80) has been specified, prefer the route with the lowest CLUSTER_LIST length.
12. Prefer the route with the lowest neighbor address.

Dynamic Capabilities

BGP Dynamic Capabilities allow the communication of a change in a BGP peer's capabilities without having to restart the peering session.

The BGP implementation is done on a per-peer basis and in such a way that dynamic capabilities are supported as long as the BGP peer supports BGP Dynamic Capabilities. Advanced Routing Suite's BGP advertises Dynamic Capabilities in the OPEN message. If a BGP peer advertises support for BGP Dynamic Capabilities in the OPEN message, then Advanced Routing Suite turns on Dynamic Capabilities. Otherwise, the dynamic capabilities for this peer will be disabled.

The Advanced Routing Suite BGP supports the following BGP Dynamic Capabilities:

- Multi-protocol Capabilities
- Graceful Restart
- Route Refresh

BGP Commands

Global Configuration Mode BGP Commands
"router bgp" ("router bgp" on page 150)

Global BGP Commands
"address-family" ("address-family" on page 77)
"bgp always-compare-med" ("bgp always-compare-med" on page 78)
"bgp as-path-loops" ("bgp as-path-loops" on page 78)
"bgp bestpath as-path ignore" ("bgp bestpath as-path ignore" on page 79)
"bgp bestpath compare-cluster-list-length" ("bgp bestpath compare-cluster-list-length" on page 80)
"bgp bestpath compare-originator-id" ("bgp bestpath compare-originator-id" on page 81)
"bgp bestpath compare-router-id" ("bgp bestpath compare-router-id" on page 82)
"bgp bestpath med confed" ("bgp bestpath med confed" on page 82)
"bgp bestpath med missing-as-worst" ("bgp bestpath med missing-as-worst" on page 83)
"bgp cluster-id" ("bgp cluster-id" on page 84)
"bgp confederation identifier" ("bgp confederation identifier" on page 85)
"bgp confederation peers" ("bgp confederation peers" on page 86)
"bgp non-leading-confeds" ("bgp non-leading-confeds" on page 86)
"bgp pass-optionable-nontrans" ("bgp pass-optionable-nontrans" on page 88)
"bgp restart-defer" ("bgp restart-defer" on page 89)
"bgp restart-delete-remnents" ("bgp restart-delete-remnents" on page 90)
"bgp restart-time" ("bgp restart-time" on page 91)
"bgp restart-timeout" ("bgp restart-timeout" on page 91)
"bgp router-id" ("bgp router-id" on page 92)
"bgp send-group-always" ("bgp send-group-always" on page 93)
"bgp tie-break-on-age" ("bgp tie-break-on-age" on page 94)
"default-metric" ("default-metric" on page 95)
"distance" ("distance" on page 96)
"distribute-list" ("distribute-list" on page 97)
"enable" ("enable" on page 98)
"maximum-routes" ("maximum-routes" on page 99)
"network" ("network" on page 146)
"preference2" ("preference2" on page 147)
"redistribute" ("redistribute" on page 148)
"timers bgp"
"trace file"
"trace flag"

**BGP Neighbor Commands**

"neighbor add-communities" ("neighbor add-communities" on page 100)
"neighbor aggregator-id" ("neighbor aggregator-id" on page 101)
"neighbor allow" ("neighbor allow" on page 102)
"neighbor as-loop" ("neighbor as-loop" on page 103)
"neighbor as-overide" ("neighbor as-overide" on page 104)
"neighbor aspath-prepend" ("neighbor aspath-prepend" on page 105)
"neighbor capability orf comm-filter" ("neighbor capability orf comm-filter" on page 105)
"neighbor capability orf extcomm-filter" ("neighbor capability orf extcomm-filter" on page 106)
"neighbor capability orf prefix-filter" ("neighbor capability orf prefix-filter" on page 107)
"neighbor cluster-id" ("neighbor cluster-id" on page 108)
"neighbor distance" ("neighbor distance" on page 109)
"neighbor dynamic" ("neighbor dynamic" on page 110)
"neighbor enable" ("neighbor enable" on page 111)
"neighbor end-of-rib" ("neighbor end-of-rib" on page 112)
"neighbor export-localpref" ("neighbor export-localpref" on page 112)
"neighbor ignore-leading-as" ("neighbor ignore-leading-as" on page 114)
"neighbor import-localpref" ("neighbor import-localpref" on page 115)
"neighbor keep" ("neighbor keep" on page 116)
"neighbor keepalives-always" ("neighbor keepalives-always" on page 117)
"neighbor local-as" ("neighbor local-as" on page 118)
"neighbor log-up-down" ("neighbor log-up-down" on page 119)
"neighbor maximum-routes" ("neighbor maximum-routes" on page 120)
"neighbor metric-out" ("neighbor metric-out" on page 122)
"neighbor multi-protocol-next-hop" ("neighbor multi-protocol-next-hop" on page 122)
"neighbor next-hop-self" ("neighbor next-hop-self" on page 123)
"neighbor keepalives-always" ("neighbor keepalives-always" on page 117)
"neighbor local-as" ("neighbor local-as" on page 118)
"neighbor log-up-down" ("neighbor log-up-down" on page 119)
"neighbor maximum-routes" ("neighbor maximum-routes" on page 120)
"neighbor metric-out" ("neighbor metric-out" on page 122)
"neighbor multi-protocol-next-hop" ("neighbor multi-protocol-next-hop" on page 122)
"neighbor next-hop-self" ("neighbor next-hop-self" on page 123)
"neighbor orf comm-list" ("neighbor orf comm-list" on page 124)
"neighbor orf extcomm-list" ("neighbor orf extcomm-list" on page 125)
"neighbor orf prefix-list" ("neighbor orf prefix-list" on page 126)
"neighbor out-delay" ("neighbor out-delay" on page 127)
"neighbor passive" ("neighbor passive" on page 128)
"neighbor password" ("neighbor password" on page 128)
"neighbor pedantic" ("neighbor pedantic" on page 129)
"neighbor peer-group" ("neighbor peer-group" on page 130)
"neighbor preference2" ("neighbor preference2" on page 132)
"neighbor receive-buffer" ("neighbor receive-buffer" on page 133)
"neighbor remote-as" ("neighbor remote-as" on page 133)
"neighbor remove-private-as" ("neighbor remove-private-as" on page 134)
"neighbor route-map" ("neighbor route-map" on page 135)
"neighbor route-reflector-client" ("neighbor route-reflector-client" on page 136)
"neighbor route-to-peer" ("neighbor route-to-peer" on page 137)
"neighbor send-buffer" ("neighbor send-buffer" on page 138)
"neighbor send-community" ("neighbor send-community" on page 139)
"neighbor soft-reconfiguration inbound" ("neighbor soft-reconfiguration inbound" on page 140)
"neighbor timers" ("neighbor timers" on page 140)
"neighbor ttl" ("neighbor ttl" on page 142)
"neighbor update-source" ("neighbor update-source" on page 142)
"neighbor use-med" ("neighbor use-med" on page 144)
"neighbor v4-gateway" ("neighbor v4-gateway" on page 145)
"neighbor version" ("neighbor version" on page 146)

Querying and Clearing Commands
"clear ip bgp" ("clear ip bgp" on page 94)
"show ip bgp" ("show ip bgp" on page 151)
"show ip bgp instance"
"show ip bgp neighbors"
"show ip bgp orf"
"show ip bgp paths"
"show ip bgp peer-group"
"show ip bgp summary"
**address-family**

**Name**
address-family - configures multi-protocol support for BGP peers

**Syntax**

```
address-family [ipv4 (unicast | multicast | vpn)]
no address-family [ipv4 (unicast | multicast | vpn)]
```

**Mode**
BGP Router Configuration

**Parameters**

- **ipv4** - configures address family mode for either IPv4
- **unicast** | **multicast** - configures address family mode for unicast
- **multicast** - configures address family mode for either multicast
- **vpn** - configures address family mode for VPN

**Description**
The `address-family` command is a mode within a BGP router configuration. It is used to configure multi-protocol support for BGP peers.

**Notes**
- The `vpn` keyword is only available if one of the following flags is used to build Advanced Routing Suite: `--enable-vri`, `--enable-vpn4`, or `--enable-mpbpg_vpn4`. See "Appendix A Build Flags" in *Installing Advanced Routing Suite NGC 2.4* for more information.

**Default**
This command is not explicitly configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example will enter address family mode for IPv4 multicast. Notice that the prompt changes to 
(config-router-af)#.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# address-family ipv4 multicast
(config-router-af)# neighbor 1.2.3.4 activate
```

**Example 2**
The following example removes all address-family configuration for IPv4 multicast.

```
(config)# router bgp 64512
(config-router-bgp)# no address-family ipv4 multicast
```
bgp always-compare-med

Name
bgp always-compare-med - specifies whether to compare when routes with differing Multi-Exit Discriminators (MEDs) are received from peers in different Autonomous Systems

Syntax
bgp always-compare-med
no bgp always-compare-med

Mode
BGP Router Configuration

Parameters
none

Description
When two routes to the same destination are received from peers in different Autonomous Systems, they can have different MEDs. The bgp always-compare-med command allows you to specify whether to compare those MEDs. When choosing between these routes, assuming that nothing else makes one preferable to the other (such as a configured policy), the values of the differing MEDs are used to choose the route to use. In this comparison, the route with the lowest MED is preferred. Routes without MEDs are treated as having the best possible MED.

Default
If bgp always-compare-med is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp always-compare-med
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example enables the comparison of MEDs among confederation paths.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp always-compare-med
(config-router-bgp)# exit
```

bgp as-path-loops

Name
bgp as-path-loops - specifies the number of times an autonomous system may appear in an AS path

Syntax
bgp as-path-loops loop-count
no bgp as-path-loops loop-count?

Mode
BGP Router Configuration

Parameters
loop-count – an integer in the range 1 to 10, inclusive

Description
The `bgp as-path-loops` command specifies the number of times this autonomous system can appear in an AS path. `loop-count` is an integer in the range 1 to 10, inclusive. The negative form of this command, `no bgp as-path-loops`, removes the configured `loop-count` value and returns this to its default value of 1.

Notes:
- Specifying a value for `loop-count` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- This command should not be used in normal operations.

Default
If `bgp as-path-loops` is not specified for an autonomous system setting, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp as-path-loops 1
```

Command History
NGC 2.2 - This command was introduced.

Examples
In this example, the router's AS number is set to 7476 and the number of loops is set to 2.

```
(config)# router bgp 7476
(config-router-bgp)# bgp as-path-loops 2
(config-router-bgp)# exit
```

**bgp bestpath as-path ignore**

Name
`bgp bestpath as-path ignore` – specifies to ignore the AS_PATH length when breaking ties between BGP routes

Syntax
```
bgp bestpath as-path ignore ignore
no bgp bestpath as-path ignore
```

Mode
BGP Router Configuration

Parameters
none
Description

Use `bgp bestpath as-path ignore` when you do not want the AS_PATH length to be considered when breaking ties between BGP routes.

Default

If `bgp bestpath as-path ignore` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp bestpath as-path ignore
```

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example Advanced Routing Suite will ignore the AS_PATH length when breaking ties

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp bestpath as-path ignore
(config-router-bgp)# exit
(config)
```

**bgp bestpath compare-cluster-list-length**

**Name**

`bgp bestpath compare-cluster-list-length` - enables breaking ties based on shortest cluster list length

**Syntax**

```
bgp bestpath compare-cluster-list-length
no bgp bestpath compare-cluster-list-length
```

**Mode**

BGP Router Configuration

**Parameters**

`none`

**Description**

Use the `bgp bestpath compare-cluster-list-length` command to break ties between routes based on shortest cluster list length.

**Default**

If `bgp bestpath compare-cluster-list-length` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp bestpath compare-cluster-list-length
```

**Command History**

NGC 2.2 - This command was introduced.
Examples
In the following example Advanced Routing Suite will consider cluster list length when breaking ties and use the route with the lowest length.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp bestpath compare-cluster-list-length
(config-router-bgp)# exit
(config)
```

**bgp bestpath compare-originator-id**

**Name**

bgp bestpath compare-originator-id - specifies to break ties based on lowest Originator ID value

**Syntax**

```
bgp bestpath compare-originator-id
no bgp bestpath compare-originator-id
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

Use the bgp bestpath compare-originator-id command to break ties between routes based the Originator ID value instead of the neighbor’s router ID. When this command is configured, the route with the lowest Originator ID will be preferred over the router ID.

**Default**

If bgp bestpath compare-originator-id is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp bestpath compare-originator-id
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example Advanced Routing Suite will consider the Originator ID when breaking ties and use the route with the lowest Originator ID.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp bestpath compare-originator-id
(config-router-bgp)# exit
(config)
```
bgp bestpath compare-router-id

**Name**

`bgp bestpath compare-router-id` - specifies whether to ignore router IDs when breaking ties between BGP routes

**Syntax**

```
bgp bestpath compare-router-id
no bgp bestpath compare-router-id
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

By default, router IDs are considered when breaking ties with BGP routes. Use the `no bgp bestpath compare-router-id` command when you do not want router IDs to be considered when breaking ties between BGP routes.

**Default**

Router IDs are compared by default. Therefore, if `bgp bestpath compare-router-id` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp bestpath compare-router-id
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example cause Advanced Routing Suite to ignore router IDs when breaking ties between BGP routes.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# no bgp bestpath compare-router-id
(config-router-bgp)# exit
(config)#
```

bgp bestpath med confed

**Name**

`bgp bestpath med confed` - specifies that confederation segments within a route’s AS_PATH should not be considered when determining the neighboring AS that a route was learned from, during MED comparison

**Syntax**

```
bgp bestpath med confed
no bgp bestpath med confed
```
**Mode**
BGP Router Configuration

**Parameters**
one

**Description**
When comparing MEDs, use `bgp bestpath med confed` to specify whether confederation segments within a route’s AS_PATH should not be considered when determining the neighboring AS that a route was learned from.

**Default**
If `bgp bestpath med confed` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp bestpath med confed
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example enables the comparison of MEDs among confederation paths.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp bestpath med confed
(config-router-bgp)# exit
(config)#
```

### bgp bestpath med missing-as-worst

**Name**

bgp bestpath med missing-as-worst - specifies whether routes received without a MED are to be considered when comparing BGP routes

**Syntax**

```
bgp bestpath med missing-as-worst
no bgp bestpath med missing-as-worst
```

**Mode**
BGP Router Configuration

**Parameters**
one

**Description**
When breaking ties between BGP routes, use the `bgp bestpath med missing-as-worst` command to specify whether routes received without a MED are to be considered to have the worst legal MED.
Default

If `bgp bestpath med missing-as-worst` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp bestpath med missing-as-worst
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example cause Advanced Routing Suite to consider routes without a MED to have the worst legal MED.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp bestpath med missing-as-worst
(config-router-bgp)# exit
(config)#
```

`bgp cluster-id`

Name

`bgp cluster-id` - specifies the route reflection cluster ID for BGP

Syntax

```
bgp cluster-id router-id
no bgp cluster-id router-id
```

Mode

BGP Router Configuration

Parameters

`router-id` - a router ID in dotted-quad format used by route reflectors to prevent route propagation loops within the cluster

Description

The `bgp cluster-id` command specifies the route reflection cluster ID for BGP. The cluster ID defaults to be the same as the router ID. If a router is to be a route reflector, then a single cluster ID should be selected and configured on all route reflectors in the cluster. If there is only one route reflector in the cluster, the cluster-id setting can be omitted because the default will suffice.

The only constraints on the choice of cluster ID are the following:

- IDs of clusters within an Autonomous System (AS) must be unique within that AS.
- The cluster ID must not be `0.0.0.0`

Default

The cluster ID value defaults to the router ID.

Command History

NGC 2.2 - This command was introduced.
**Examples**
The following example configures a cluster ID of 1.2.3.4 for AS 64512.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# bgp cluster-id 1.2.3.4
(config-router-bgp)# exit
(config)#
```

**bgp confederation identifier**

**Name**

`bgp confederation identifier` - configures the BGP router to be a member of a BGP confederation

**Syntax**

`bgp confederation identifier confed_id`

`no bgp confederation identifier confed_id`

**Mode**

BGP Router Configuration

**Parameters**

`confed_id` - an integer between 1 and 65535, inclusive

**Description**

A BGP router can be configured to be a member of a BGP confederation where the autonomous system is subdivided into several confederation AS’s. When configured as a confederation member, this router will represent itself as the configured AS number to confederation peers and as the configured confederation identifier to non-confederation peers.

Use the negative form of this command, `no bgp confederation identifier`, to remove the configured confederation peer identifier.

**Default**

No BGP confederation identifiers are configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the BGP router to be a member of BGP confederation 65000.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp confederation identifier 65000
(config-router-bgp)# exit
(config)#
```
**bgp confederation peers**

**Name**

`bgp confederation peers` - configures a BGP group to be part of a BGP confederation

**Syntax**

```
bgp confederation peers as_number {1, n}
no bgp confederation peers as_number {1, n}
```

**Mode**

BGP Router Configuration

**Parameters**

- `as_number` - an autonomous system number of a group of peers that should be designated as confederation peers
- `{1, n}` - When specified, the `bgp confederation peers` command must include at least one `as_number`

**Description**

The `bgp confederation peers` command configures a BGP group to be part of a BGP confederation. When this command is present, Advanced Routing Suite will treat all members of that group as confederation peers.

**Default**

Confederation peers are not configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures peers in AS number 65535 and AS number 42 to be confederation peers.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp confederation peers 65535 42
(config-router-bgp)# exit
(config)#
```

**bgp non-leading-confeds**

**Name**

`bgp non-leading-confeds` - specifies the action to take when malformed AS_PATHs containing non-prefixed confederation segments are received

**Syntax**

```
bgp non-leading-confeds [ notify | ignore | drop ]
no bgp non-leading-confeds [ notify | ignore | drop ]
```
Mode
BGP Router Configuration

Parameters

**notify** - specifies that receipt of non-prefixied confederation segments will cause BGP to send a notification message

**ignore** - specifies that routes containing non-prefixied confederation segments will be logged and ignored

**drop** - specifies that routes containing non-prefixied confederation segments will be logged and discarded

Description

"Buggy" routers on the Internet will sometimes illegally advertise AS_PATHs containing confederation segments outside of a confederation boundary. If these routes propagate beyond the confederation boundary edge, they will cause the peering session of any router that does not accept confederation segments from non-confederation peers to drop the peering session and thus disrupt service.

**drop** causes BGP to discard AS_PATHs that are received from BGP peers where there are BGP Confederation AS_PATH segments (AS_CONFED_SEQUENCE, AS_CONFED_SET) occurring anywhere other than at the left-hand side of the AS_PATH.

**ignore** causes BGP to ignore AS_PATHs that are received from BGP peers where there are BGP Confederation AS_PATH segments (AS_CONFED_SEQUENCE, AS_CONFED_SET) occurring anywhere other than at the left-hand side of the AS_PATH. This option will cause routes containing non-prefixied confederation segments to be logged and stored in the RIB with a preference or distance of -1.

**notify** causes BGP to send a notification upon receipt of BGP peers where there are BGP Confederation AS_PATH segments (AS_CONFED_SEQUENCE, AS_CONFED_SET) occurring anywhere other than at the left-hand side of the AS_PATH.

The negative form of this command, **no bgp non-leading-confeds**, removes the user-defined configuration and returns this to its default value of **notify**. **Note:** Specifying a parameter for this command in the **no** form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If **bgp non-leading-confeds** is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp non-leading-confeds notify
```

Command History

NGC 2.2 - This command was introduced.

Examples

**Example 1**
The following example causes Advanced Routing Suite to log and ignore non-prefixied confederation segments.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp non-leading-confeds ignore
(config-router-bgp)# exit
(config)#
```

**Example 2**
The following example removes the previously configured ignore command and returns this to its default value of **notify**.

```
(config)# router bgp 65535
```
(config-router-bgp)# no bgp non-leading-confeds
(config-router-bgp)# exit
(config)#

bgp open-on-accept

**Name**

bgp open-on-accept - specifies that Advanced Routing Suite will send the Open message when the TCP connection has completed for configured peers

**Syntax**

```plaintext
bgp open-on-accept
no bgp open-on-accept
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

When the `bgp open-on-accept` command is configured, Advanced Routing Suite will immediately send the Open message when the TCP connection has completed for configured peers.

**Default**

If `bgp open-on-accept` is not specified, it is the same as if the user had specified the following:

```plaintext
(config-router-bgp)# no bgp open-on-accept
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example enables the open-on-accept feature.

```plaintext
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp open-on-accept
(config-router-bgp)# exit
(config)#
```

bgp pass-optional-nontrans

**Name**

bgp pass-optional-nontrans - specifies whether to propagate unrecognized, optional, non-transitive attributes

**Syntax**

```plaintext
bgp pass-optional-nontrans
```
no bgp pass-optinal-nontrans

Mode
BGP Router Configuration

Parameters
none

Description
Use the `bgp pass-optinal-nontrans` command to specify whether to propagate unrecognized, optional, non-transitive attributes.

Default
If `bgp pass-optinal-nontrans` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp pass-optinal-nontrans
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example enables the pass-optinal-nontrans feature.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp pass-optinal-nontrans
(config-router-bgp)# exit
(config)#
```

**bgp restart-defer**

Name
`bgp restart-defer` - configures the time to defer route selection after gracefully restarting

Syntax
```
bgp restart-defer time-seconds
no bgp restart-defer time-seconds?
```

Mode
BGP Router Configuration

Parameters

`time-seconds` - time in seconds between 0 and 4,294,967,295, inclusive

Description
Use the `bgp restart-defer` command to configure the time in seconds to defer route selection after gracefully restarting. This is the maximum time to wait for all peers to come back up and send End-Of-RIB messages. The negative form of this command, `no bgp restart-defer`, removes the configured `time-seconds` value and returns this to its default value of 180 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
**Default**

If `bgp restart-defer` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp restart-defer 180
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the restart-defer time to be 150 seconds.

```
(config)# router bgp 2222
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# bgp restart-defer 150
(config-router-bgp)# exit
```

### bgp restart-delete-remnants

**Name**

`bgp restart-delete-remnants` - turns on BGP’s deletion of kernel remnant routes

**Syntax**

```
bgp restart-delete-remnants
no bgp restart-delete-remnants
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

The `bgp restart-delete-remnants` command turns on BGP's deletion of kernel remnant routes when performing a graceful restart after all expected End-of-RIB messages have been received or the restart-defer timer expires. This is used to remove stale routing information that had been retained by the graceful restart procedure.

The negative form of this command, `no bgp restart-delete-remnants`, turns this feature off. In this case, stale routing information will be retained after a graceful restart until the kernel remnant hold timer expires. (See the "kernel remnant-holdtime" command.)

**Default**

If `bgp restart-delete-remnants` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp restart-delete-remnants
```

**Command History**

NGC 2.3 - This command was introduced.

**Examples**

The following example turns off BGP’s deletion of kernel remnant routes.

```
(config)# router bgp 2222
```
bgp restart-time

Name
bgp restart-time - configures the time advertised within the graceful restart capability when opening a peering session

Syntax
bgp restart-time time-seconds
no bgp restart-time time-seconds?

Mode
BGP Router Configuration

Parameters
time-seconds - the time in seconds between 0 and 4,294,967,295

Description
Use the bgp restart-time command to configure the time advertised within the graceful restart capability when opening a peering session. This is the amount of time it takes to restart within the configured peers. The negative form of this command, no bgp restart-time, removes the configured time-seconds value and returns this to its default value of 180 seconds. Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If bgp restart-time is not specified, it is the same as if the user had specified the following:

(config-router-bgp)# bgp restart-time 180

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the restart-time to be 150 seconds.

(config)# router bgp 2222
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp restart-time 150
(config-router-bgp)# exit
(config)#

bgp restart-timeout

Name
bgp restart-timeout - configures the time to wait for gracefully restarting peers to come back up
Syntax

bgp restart-timeout time-seconds
no bgp restart-timeout time-seconds?

Mode
BGP Router Configuration

Parameters

time-seconds - the time in seconds between 0 and 4,294,967,295

Description

Use the `bgp restart-timeout` command to configure the time in seconds to wait for gracefully restarting peers to come back up. This overrides the value that the peer advertised in the graceful start capability. Once this timeout is reached, Advanced Routing Suite will delete all routes learned from the restarting peer. The negative form of this command, `no bgp restart-timeout`, removes the configured `time-seconds` value and returns this to its default value of 180 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `bgp restart-timeout` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# bgp restart-timeout 180
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the restart-timeout to be 150 seconds.

```
(config)# router bgp 2222
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp restart-timeout 150
(config-router-bgp)# exit
```

**bgp router-id**

**Name**

bgp router-id - configures a BGP-specific router ID

**Syntax**

```
bgp router-id router_id
no bgp router-id router_id
```

**Mode**

BGP Router Configuration

**Parameters**

`router_id` - a 32-bit integer in dotted-quad notation to be used as the BGP router ID
**Description**

Use the `bgp router-id` command to specify a Router ID value that overrides the globally configured Router ID.

**Default**

BGP defaults to the globally configured Router ID.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the BGP Router ID to be 1.2.3.4.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# exit
(config)#
```

**bgp send-group-always**

**Name**

`bgp send-group-always` - an internal debugging option that causes BGP packets to be sent via the code optimized for sending to multiple peers simultaneously, when it makes sense to do so.

**Syntax**

```
bgp send-group-always
no bgp send-group-always
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

`bgp send-group-always` is an internal debugging option. By specifying this command, BGP packets will be sent via the code optimized for sending to multiple peers simultaneously, when it makes sense to do so. Use the negative form of this command, `no bgp send-group-always`, to turn this setting off.

**Default**

If `bgp send-group-always` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no bgp send-group-always
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example causes BGP packets to be sent via the code optimized for sending to multiple peers simultaneously.

```
(config)# router bgp 65000
```
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp send-group-always
(config-router-bgp)# exit
(config)#

**bgp tie-break-on-age**

**Name**
bgp tie-break-on-age - specifies that the age of routes should be compared when breaking ties between BGP routes

**Syntax**
bgp tie-break-on-age
no bgp tie-break-on-age

**Mode**
BGP Router Configuration

**Parameters**
none

**Description**
When breaking ties between BGP routes, bgp tie-break-on-age can be used to specify that the age of the routes should be compared. If this is configured, then older routes are favored over newer routes.

**Default**
If bgp tie-break-on-age is not specified, it is the same as if the user had specified the following:
(config-router-bgp)# no bgp tie-break-on-age

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example causes BGP to break ties in favor of older routes.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# bgp tie-break-on-age
(config-router-bgp)# exit
(config)#
```

**clear ip bgp**

**Name**
clear ip bgp - resets BGP peering sessions and sends route refresh requests

**Syntax**
clear ip bgp [peer | *] [soft]?
Mode
User Execution

Parameters
[peer | *] - specify the IPv4 address of a single BGP peer, or specify "*" to clear all peers
soft - optionally specify to send a Route Refresh message (if supported by the peer)

Description
Use the clear ip bgp command to reset BGP peering sessions or to send a Route Refresh request.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example specifies to clear the BGP peer 1.2.3.4.

    > clear ip bgp 1.2.3.4

Example 2
The following example specifies to clear all BGP peers and send a Route Refresh message.

    > clear ip bgp * soft

default-metric

Name
default-metric - causes a BGP MED to be set on routes when they are advertised to peers

Syntax
default-metric metric
default-metric?

Mode
BGP Router Configuration

Parameters
metric - a 32-bit value ranging from 0 to 4,294,967,295, inclusive

Description
The default-metric command causes a BGP MED to be set on routes when they are advertised to peers. This value applies to all BGP peers. It can be overridden on a per-peer or per-group basis. The negative form of this command, no default-metric, removes the configured value. Note: Specifying a value for metric in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
The default-metric command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.
Examples

Example 1
The following example causes Advanced Routing Suite to advertise a MED value of 50 to its BGP peers.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# default-metric 50
(config-router-bgp)# exit
(config)#
```

Example 2
The following example removes the previously configured default-metric value on the router.

```
(config)# router bgp 65535
(config-router-bgp)# no default-metric 50
(config-router-bgp)# exit
(config)#
```

distance

Name
distance - specifies the way that active routes learned from BGP will be selected, compared to other protocols

Syntax

distance dist
no distance dist?

Mode
BGP Router Configuration

Parameters

dist - an integer between 0 and 255, inclusive

Description
The distance command specifies how active routes that are learned from BGP will be selected, compared to other protocols. When a route has been learned from more than one protocol, the active route will be selected from the protocol with the lowest distance (or preference).

The negative form of this command, no distance, removes the configured value and returns this to its default value of 170. Note: Specifying a value for dist in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If distance is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# distance 170
```

Command History
NGC 2.2 - This command was introduced.
Examples

Example 1
The following example configures the global BGP distance (preference) to be 140.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# distance 140
(config-router-bgp)# exit
(config)#
```

Example 2
The following example deletes the globally configured BGP distance (preference) and reverts to the default value of 170.

```
(config)# router bgp 65535
(config-router-bgp)# no distance 140
(config-router-bgp)# exit
(config)#
```

distribute-list

Name
distribute-list – provides policy filtering for BGP routes

Syntax
```
distribute-list access_list_name [ in | out [protocol]? ]
no distribute-list access_list_name [ in | out [protocol]? ]
```

Mode
BGP Router Configuration

Parameters
access_list_name – the name of an access list

in | out – specifies whether the route applies to imported (in) or exported (out) routes

protocol – for exported routes (out), optionally specify a redistributed protocol to filter. Valid values include: aggregate, bgp, kernel, ospf, ospf-ase, rip, and static.

Description
The `distribute-list` command provides a policy filtering mechanism for BGP routes. If the distribute list configured in this command is specified with the `in` keyword, then the filter will apply to all imported routes (that is to say, routes learned from BGP neighbors). If the distribute list configured in this command is specified with the `out` keyword, then the filter applies to exported routes (that is to say, routes announced to BGP neighbors).

To delete a configured distribute list, use the negative form of the command. **Note:** All arguments of the original command must be supplied in order for the entry to be deleted.
distribute-list in

The `distribute-list in` command configures a BGP import policy (in other words, a policy for BGP to apply to incoming updates). If an inbound `distribute-list` is configured, then each route received from other BGP neighbors will be evaluated against this inbound list. If the route matches the criteria specified by the inbound list, then the route is imported. If the route does not match the inbound list criteria, then the route is rejected. A route matches the inbound list criteria if it is permitted when evaluated against the referenced access list.

distribute-list out

The `distribute-list out` command is used to configure export policy for BGP routes. Outbound distribute lists look similar to inbound distribute lists, except that they allow you to optionally specify a protocol, or more correctly, a route source. This allows you to filter exported routes based on whether they were learned from RIP, OSPF, OSPF-ASE or were aggregate, static, or kernel routes. At most one `distribute-list out` command that does not reference a protocol can be specified. If an outbound distribute list is configured without referencing a protocol, then when a route is being considered for export to BGP neighbors, it must be permitted by the access list referenced in the distribute list. If instead an outbound distribute list is configured that references a protocol, and if the route being considered for export originated from the referenced protocol, then the route must be permitted by the access list referenced in the distribute list. If more of both types of lists are present, then the route need only be permitted by the access list referenced by one of the outbound distribute lists.

It should be noted that an outbound distribute list has no effect in the absence of a "redistribute" ("redistribute" on page 148) command. Outbound distribute lists can be thought of as a way to further refine the export policy expressed with a "redistribute" ("redistribute" on page 148) command.

The negative form of this command, `no distribute-list`, removes the configured list. You must specify either in or out in the negative form. For outbound distribute lists, specifying a protocol is optional. If the protocol is not specified, then all distribute lists referencing the specified access list will be deleted.

Default

Distribute lists are not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures two distribute lists. The first distribute list defines import policy using access list "alist1". The second distribute list defines export policy for static routes.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# distribute-list alist1 in
(config-router-bgp)# distribute-list alist2 out static
(config-router-bgp)# exit
(config)#
```

See Also

"redistribute" ("redistribute" on page 148)

enable

Name

enable - enables BGP
**Syntax**

```
enable
no enable
```

**Mode**

BGP Router Configuration

**Parameters**

none

**Description**

The `enable` command enables BGP configuration on a router. The negative form of this command, `no enable`, disables BGP. This command is useful when you want BGP turned "on" on a router, but you want your BGP router configuration details to be turned "off."

**Default**

If `enable` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# enable
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example disables BGP on the router. Note that BGP will still be turned "on" on AS 65535, but configuration state will be turned off.

```
(config)# router bgp 65535
(config-router-bgp)# no enable
(config-router-bgp)# exit
(config)#
```

---

**maximum-routes**

**Name**

`maximum-routes` - specifies the maximum number of routes that BGP will accept for installation into the Advanced Routing Suite routing information base (RIB)

**Syntax**

```
maximum-routes number
no maximum-routes number?
```

**Mode**

BGP Router Configuration

**Parameters**

`number` - an integer between 1 and 4,294,967,295, inclusive

**Description**

The `maximum-routes` command specifies the maximum number of routes that BGP will accept for installation into the Advanced Routing Suite RIB.
Default

If `maximum-routes` is not specified, then BGP will accept an unlimited number of routes for installation into the Advanced Routing Suite RIB. The negative form of this command, `no maximum-routes`, removes the configured value and allows an unlimited number of routes. **Note:** Specifying a value for `number` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, Advanced Routing Suite will install a maximum of 50 routes learned from BGP into the routing table.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# maximum-routes 50
(config-router-bgp)# exit
(config)#
```

eighbor add-communities

**Name**

`neighbor add-communities` - adds a community to BGP

**Syntax**

```
neighbor ip_address add-communities id
no neighbor ip_address add-communities id
```

**Mode**

BGP Router Configuration

**Parameters**

`ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format.

`id` - the community set ID to be added

**Description**

Use the `neighbor add-communities` command to add a BGP community to this peer. BGP communities are configured with the `ip community-set` command in Global Configuration mode. The `id` value corresponds with the ID of the community set.

**Default**

Community sets are not added or configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example causes the communities described in "comm1" to be sent in BGP updates to peer 1.2.3.4.

```
(config)# router bgp 60004
```
neighbor aggregator-id

Name
neighbor aggregator-id - specifies whether the router-id in the aggregator attribute should be 0

Syntax
neighbor ip_address aggregator-id
no neighbor ip_address aggregator-id

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

Description
When Advanced Routing Suite aggregates a route, the aggregator ID attribute, along with the Router ID are sent. neighbor aggregator-id allows you to specify whether the Router ID in the aggregator attribute should be 0, instead of the ID of the router.

The negative of this command, no neighbor aggregator-id, specifies that the router ID in the aggregator attribute should be the ID of the router.

Default
If neighbor aggregator-id is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] aggregator-id
```

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example causes BGP to set the value of the aggregator ID to 0.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 add-communities comm1
(config-router-bgp)# exit
(config)#
```

Example 2
The following example causes BGP to set the value of the aggregator ID to the router ID.

```
(config-router-bgp)# neighbor 1.2.3.4 aggregator-id
(config-router-bgp)# exit
(config-router-bgp)#
```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# no neighbor 1.2.3.4 aggregator-id
(config-router-bgp)# exit
(config)#

**neighbor allow**

**Name**

`neighbor allow` - permits peer connections from addresses in the specified access-list

**Syntax**

```
neighbor group-name allow access_list remote-as as-number
no neighbor group-name allow access_list
```

**Mode**

BGP Router Configuration

**Parameters**

- `group-name` - the name of a BGP peer group specified as a string of characters
- `access_list` - the name of the access list from which peer connections will be made
- `remote-as as_num` - the autonomous system (AS) number of a BGP peer. This can be an integer from 1 to 65535, inclusive.

**Description**

The `neighbor allow` command permits peer connections from addresses in the specified access list. Access lists are configured with the `ip access-list` command in Global Configuration mode. The `access_list` value corresponds with the ID of the access list.

**Note:** You must specify the remote Autonomous System again in this command, even if it was previously specified in the `neighbor` command.

**Default**

The allow command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example allows incoming peering connections from those peers specified in access-list-1.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor group1 allow access-list-1 remote-as 5
(config-router-bgp)# exit
```
Example 2

The following example will cease allowing incoming peering connections from access_list_1.

```
(config)# router bgp 65535
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# no neighbor 1.2.3.4 allow access-list-1 remote-as 5
(config-router-bgp)# exit
(config)#
```

See Also

Chapter 25 Access Lists

neighbor as-loop

Name

`neighbor as-loop` - specifies the number of times an autonomous system may appear in an AS path

Syntax

```
neighbor [ ip_address | group-name ] as-loop loop-count
no neighbor [ ip_address | group-name ] as-loop loop-count?
```

Mode

BGP Router Configuration

Parameters

- `ip_address` - the address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `loop-count` - an integer in the range 1 to 10, inclusive

Description

The `neighbor as-loop` command specifies the number of times this autonomous system can appear in a peer or peer group AS path. `loop-count` is an integer in the range 1 to 10, inclusive. The negative form of this command, `no neighbor as-loop`, removes the configured `loop-count` value and returns this to its default value of 1.

Notes:

- Specifying a value for `loop-count` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- This command should not be used in normal operations.

Default

If `neighbor as-loop` is not specified for an autonomous system setting, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] as-loop 1
```
Command History
NGC 2.2 - This command was introduced.

Examples
In this example, the number of loops for peer 1.2.3.4 is set to 2.

```
(config)# router bgp 7476
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 as-loop 2
(config-router-bgp)# exit
(config)#
```

**neighbor as-override**

**Name**
neighbor as-override - configures all occurrences of a peer’s AS to be replaced with one from an export

**Syntax**
```
neighbor [ ip_address | group-name ] as-override
no neighbor [ ip_address | group-name ] as-override
```

**Mode**
BGP Router Configuration

**Parameters**
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**
The neighbor as-override command specifies that all occurrences of a peer’s AS should be replaced with our address. The negative form of this command, no neighbor as-override, removes this configuration.

**Default**
If neighbor as-override is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ ip_address | group-name ] as-override
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example turns on the AS Override feature.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 as-override
```
neighbor aspath-prepend

Name

neighbor aspath-prepend - specifies the number of times that this router will prepend its AS number to a route's AS Path when it sends the route to an external peer.

Syntax

neighbor ip_address aspath-prepend num
no neighbor ip_address aspath-prepend num

Mode

BGP Router Configuration

Parameters

ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
num - an integer between 1 and 25, inclusive

Description

Use the neighbor aspath-prepend command to configure the number of times that this router will prepend its AS number to a route's AS Path when it sends the route to an external peer. Larger values are typically used to bias upstream peers' route selection. Most routers will prefer routes with shorter AS Path. Using this command, the AS Path that this router sends can be artificially lengthened.

Default

If neighbor aspath-prepend is not specified, it is the same as if the user had specified the following:

(config-router-bgp)# neighbor [ip_address | group-name] aspath-prepend 1

Note: The BGP protocol specifies you must add your local AS number to the AS_PATH attribute. The aspath-prepend option lets you do it more than once.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, the router is configured to prepend its AS number to a route's AS Path five times.

(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 10.1.1.1 remote-as 5
(config-router-bgp)# neighbor 10.1.1.1 aspath-prepend 5
(config-router-bgp)# exit
(config)#

neighbor capability orf comm-filter

Name

neighbor capability orf comm-filter - specifies whether the ORF capability for community filtering is to be sent to the BGP peer or group, received from the BGP peer or group, or both.
Syntax

```plaintext
neighbor [ip_address | group-name] capability orf
  comm-filter [send | receive | both]

no neighbor [ip_address | group-name] capability orf
  comm-filter [send | receive | both]
```

Mode
BGP Router Configuration

Parameters

- `ip_address` - the IPv4 address of a BGP peer specified in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `send | receive | both` - specifies the action to take when filtering communities

Description
The `neighbor capability orf comm-filter` commands specifies whether the ORF capability for community filtering is to be sent from a BGP peer or group, accepted on a BGP peer or group, or both. The negative form of this command, `no neighbor capability orf comm-filter`, removes the capability.

Default
ORF capabilities are not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that BGP peer 4.3.2.1 will send the ORF capability for community filtering.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf comm-filter send
```

neighbor capability orf extcomm-filter

Name
neighbor capability orf extcomm-filter - specifies whether the ORF capability for extended community filtering is to be sent to the BGP peer or group, received from the BGP peer or group, or both

Syntax

```plaintext
neighbor [ip_address | group-name] capability orf
  extcomm-filter [send | receive | both]

no neighbor [ip_address | group-name] capability orf
  extcomm-filter [send | receive | both]
```

Mode
BGP Router Configuration
Parameters

`ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

`group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

`send | receive | both` - specifies the action to take when filtering extended communities

Description

The `neighbor capability orf extcomm-filter` commands specifies whether the ORF capability for extended community filtering is to be sent from a BGP peer or group, accepted on a BGP peer or group, or both. The negative form of this command, `no neighbor capability orf extcomm-filter`, removes the capability.

**Note:** When configured in BGP Router Configuration mode, the address family sent will be IPv4 unicast.

Default

ORF capabilities are not explicitly configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example specifies that BGP peer 4.3.2.1 will send the ORF capability for extended community filtering.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf extcomm-filter send
```

neighbor capability orf prefix-filter

Name

`neighbor capability orf prefix-filter` - specifies whether the ORF capability for prefix filtering is to be sent from and/or received on the BGP peer or group.

Syntax

```
neighbor [ip_address | group-name] capability orf prefix-filter [send | receive | both]
no neighbor [ip_address | group-name] capability orf prefix-filter [send | receive | both]
```

Mode

BGP Router Configuration

Parameters

`ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

`group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

`send | receive | both` - specifies the action to take when filtering prefixes
Description

The `neighbor capability orf prefix-filter` commands specifies whether the ORF capability for prefix filtering is to be sent from a BGP peer or group, accepted on a BGP peer or group, or both. The negative form of this command, `no neighbor capability orf prefix-filter`, removes the capability.

Default

ORF capabilities are not explicitly configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example specifies that BGP peer 4.3.2.1 will send the ORF capability for prefix filtering.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf prefix-filter send
```

neighbor cluster-id

Name

`neighbor cluster-id` - specifies the route reflection cluster ID for a peer or peer group in BGP

Syntax

```
neighbor [ip_address | group-name] cluster-id router-id
no neighbor [ip_address | group-name] cluster-id router-id
```

Mode

BGP Router Configuration

Parameters

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `router-id` - a router ID in dotted-quad format used by route reflectors to prevent route propagation loops within the cluster

Description

The `neighbor cluster-id` command specifies the route reflection cluster ID for a BGP peer or peer group. The cluster ID defaults to be the same as the router ID. If a router is to be a route reflector, then a single cluster ID should be selected and configured on all route reflectors in the cluster. If there is only one route reflector in the cluster, the cluster-id setting can be omitted because the default will suffice.

The only constraints on the choice of cluster ID are the following:

- IDs of clusters within an Autonomous System (AS) must be unique within that AS.
- The cluster ID must not be 0.0.0.0
Choosing the cluster ID to be the router ID of one router in the cluster will always fulfill these criteria. If there is only one route reflector in the cluster, the "cluster ID" setting can be omitted, because the default will suffice.

When a route is received, if it does not have all of the local cluster IDs, then it is accepted. Additionally, when a route is reflected, all of the locally configured cluster IDs that are not already present on the route are prepended to the CLUSTER_LIST.

Notes:
- If configuration results in a change in the set of cluster IDs on the box, then the "clear ip bgp" ("clear ip bgp" on page 94) command is necessary to get the change to effect previous learned routes/announcements.
- You must specify either a remote AS or a peer group before configuring a cluster ID.

Default
The cluster ID value defaults to the router ID.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a cluster ID of 2.2.2.2 for peer 4.3.2.1.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 cluster ID 2.2.2.2
(config-router-bgp)# exit
(config)#
```

neighbor distance

Name
neighbor distance - specifies the way that active routes learned from BGP will be selected, compared to other protocols, within a BGP peer

Syntax
```
neighbor [ ip_address | group-name ] distance dist
no neighbor [ ip_address | group-name ] distance dist
```

Mode
BGP Router Configuration

Parameters
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `dist` - an integer between 0 and 255, inclusive

Description
The `neighbor distance` command specifies how active routes within the specified BGP peer that are learned from BGP will be selected, compared to other protocols. When a route has been learned from more
than one protocol, the active route will be selected from the protocol with the lowest distance (or preference).

The negative form of this command, no neighbor distance, removes the configured value and returns this to its default value of 170.

Default
If neighbor distance is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] distance 170
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the global BGP distance (or preference) to be 140 and the BGP distance (or preference) for peer 1.2.3.4 to be 80.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# distance 140
(config-router-bgp)# neighbor 1.1.1.1 remote-as 5
(config-router-bgp)# neighbor 1.1.1.1 distance 80
(config-router-bgp)# exit
(config)#
```

```
neighbor dynamic
```

Name
neighbor dynamic - activates the dynamic capability for this peer

Syntax
```
neighbor [ip_address | group-name] dynamic
no neighbor [ip_address | group-name] dynamic
```

Mode
BGP Router Configuration

Parameters
- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- **group-name** - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
Use the neighbor dynamic to turn on dynamic capabilities for this peer. The negative form of this command, no neighbor dynamic, turns the dynamic capability off for a peer.

Default
If neighbor dynamic is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] dynamic
```
neighbor enable

Name
neighbor enable – enables a BGP peer or peer group

Syntax
neighbor [ip_address | group-name] enable
do neighbor [ip_address | group-name] enable

Mode
BGP Router Configuration

Parameters
ip_address – the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
Use the neighbor enable command to explicitly enable a BGP peer or peer group. Use the negative form of this command, no neighbor enable, to explicitly disable a peering session. Configuration will still be retained.

Default
If neighbor enable is not specified, it is the same as if the user had specified the following:

(config-router-bgp)# neighbor [ip_address | group-name] enable

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns on the dynamic capability feature for 1.2.3.4.

(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 dynamic
(config-router-bgp)# exit

The following example enables BGP for peer 1.2.3.4.

(config)# router bgp 65535
(config-router-bgp)# neighbor 1.2.3.4 enable-peer
(config-router-bgp)# exit
(config)#
neighbor end-of-rib

**Name**
neighbor end-of-rib - activates the End-of-RIB capability for a peer

**Syntax**

```plaintext
neighbor [ip_address | group-name] end-of-rib
no neighbor [ip_address | group-name] end-of-rib
```

**Mode**
BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**
Use the `neighbor end-of-rib` command to configure the End of RIB capability for a peer. This is a special form of Graceful Restart that sends End-Of-RIB messages, but does not perform graceful restart operations.

**Default**
If `neighbor end-of-rib` is not specified, it is the same as if the user had specified the following:

```plaintext
(config-router-bgp)# no neighbor [ip_address | group-name] end-of-rib
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the End-of-RIB option for 1.2.3.4.

```plaintext
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 end-of-rib
(config-router-bgp)# exit
```

neighbor export-localpref

**Name**
neighbor export-localpref - specifies the Local Pref value that is sent in the BGP update packet when advertising a route to an internal or confederation peer

**Syntax**

```plaintext
neighbor [ip_address | group-name] export-localpref
local_pref
no neighbor [ip_address | group-name] export-localpref
local_pref
```

```plaintext
local_pref?
```
**Mode**
BGP Router Configuration

**Parameters**

- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- **group-name** - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- **local_pref** - an integer from 0 to 4,294,967,295, inclusive

**Description**

Use the `neighbor export-localpref` command to configure the Local_Pref value that is sent in the BGP Update packet when advertising a route to an internal or confederation peer. This value overrides the calculated Local_Pref on the route. BGP will export the route with the calculated Local_Pref value.

The negative form of this command, `no neighbor export-localpref`, removes the configured `local_pref` value and returns this to its default value of 100. **Note:** Specifying a value for `local_pref` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `neighbor export-localpref` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] export-localpref 100
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the export Local_Pref value to be 200.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 1.1.1.1 remote-as 5
(config-router-bgp)# neighbor 1.1.1.1 export-localpref 200
(config-router-bgp)# exit
(config)#
```

**neighbor graceful-restart**

**Name**

`neighbor graceful-restart` - configures the Graceful Restart capability for the family/sub address family pair

**Syntax**

```
neighbor [ip_address | group-name] graceful-restart
no neighbor [ip_address | group-name] graceful-restart
```

**Mode**

Address Family Configuration

**Parameters**

- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description

Use restart to turn on or off the Graceful Restart capability for the family/sub address family pair.

**Note:** For graceful restart to be useful beyond IPv4 unicast, the corresponding multi-protocol capability must also be in effect.

Default

If neighbor graceful-restart is not specified, it is the same as if the user had specified the following:

```
(config-router-af)# no neighbor [ip_address | group-name] graceful-restart
```

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example turns graceful-restart on for IPv4 multicast.

```
(config)# router bgp 61111
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# address-family ipv4 multicast
(config-router-af)# neighbor 1.2.3.4 graceful-restart
(config-router-af)# exit
(config)#
```

neighbor ignore-leading-as

Name

neighbor ignore-leading-as - directs Advanced Routing Suite to keep route server routes

Syntax

```
neighbor [ip_address | group-name] ignore-leading-as
no neighbor [ip_address | group-name] ignore-leading-as
```

Mode

BGP Router Configuration

Parameters

- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- **group-name** - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description

Some routers are capable of propagating routes without appending their own autonomous system number to the AS Path. By default, Advanced Routing Suite will drop such routes. The neighbor ignore-
leading-as command allows Advanced Routing Suite to keep these routes. This command should be used only if there is no doubt that these peers are route servers and not normal routers.

**Default**

If `neighbor ignore-leading-as` is not specified, it is the same as if the user had specified the following:

```
    (config-router-bgp)# no neighbor [ip_address | group-name] ignore-leading-as
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example enables the ignore-leading-as feature.

```
    (config)# router bgp 65535
    (config-router-bgp)# bgp router-id 1.2.3.4
    (config-router-bgp)# neighbor 1.1.1.1 remote-as 5
    (config-router-bgp)# neighbor 1.1.1.1 ignore-leading-as
    (config-router-bgp)# exit
    (config)#
```

**neighbor import-localpref**

**Name**

`neighbor import-localpref` - specifies the `Local_Pref` value that is received in the BGP Update packet when advertising a route to an external or confederation peer

**Syntax**

```
    neighbor [ip_address | group-name] import-localpref
    local_pref
    no neighbor [ip_address | group-name] import-localpref
    local_pref?
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `local_pref` - an integer from 0 to 4,294,967,295, inclusive

**Description**

Use the `neighbor import-localpref` command to configure the `Local_Pref` value that is sent in the BGP Update packet when advertising a route to an external or confederation peer. This value overrides the calculated `Local_Pref` on the route. BGP will import the route with the calculated `Local_Pref` value.

The negative form of this command, `no neighbor import-localpref`, removes the configured `local_pref` value and returns this to its default value of 100. **Note:** Specifying a value for `local_pref` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
Default
If `neighbor import-localpref` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] import-localpref 100
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the import Local_Pref value to be 200.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 1.1.1.1 remote-as 5
(config-router-bgp)# neighbor 1.1.1.1 import-localpref 200
(config-router-bgp)# exit
(config)#
```

**neighbor keep**

**Name**

`neighbor keep` - specifies the action to take when routes are received containing a router’s own autonomous system (AS) number

**Syntax**

```
neighbor [ip_address | group-name] keep [ normal | all | none ]

no neighbor [ip_address | group-name] keep [ normal | all | none ]
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `normal` - discards routes that contain a router’s own autonomous system number
- `all` - retains routes learned from a peer even if the routes’ AS paths contain the router's own AS number
- `none` - discards routes that have failed import policy

**Description**

Use `neighbor keep` to specify whether to keep routes containing a router’s own autonomous system number or routes that have failed import policy.

- `normal` causes Advanced Routing Suite to discard routes that contain a router's own autonomous system number. Also by default, Advanced Routing Suite will retain routes that have failed import policy so that these routes may be re-examined when import policy is changed during reconfiguration.
- `all` causes Advanced Routing Suite to retain all routes that contain the router's own autonomous system number. In combination with the "loops" command in the autonomous system command, this can allow a healing of a partitioned AS via the Internet.
none causes Advanced Routing Suite to discard all routes that have failed import policy. This can result in a substantial saving in space in the router. In order to re-acquire discarded routes, the "clear ip bgp" ("clear ip bgp" on page 94) command should be issued to either manually flap peering sessions or issue a Route Refresh request (with the "soft" keyword).

Default

If neighbor keep is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] keep normal
```

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example causes Advanced Routing Suite to keep routes learned from this peer even if they container our own AS number.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 keep all
(config-router-bgp)# exit
(config)#
```

Example 2

The following example causes Advanced Routing Suite to revert to the value of "normal" for keep.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# no neighbor 1.2.3.4. keep all
(config-router-bgp)# exit
(config)#
```

neighbor keepalives-always

Name

neighbor keepalives-always - causes Advanced Routing Suite to always send keepalives

Syntax

```
neighbor [ip_address | group-name] keepalives-always
no neighbor [ip_address | group-name] keepalives-always
```

Mode

BGP Router Configuration

Parameters

ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
**Description**

The `neighbor keepalives-always` command causes Advanced Routing Suite to always send keepalives, even when an update may have correctly substituted for one. This command allows interoperability with routers that do not completely obey the protocol specifications on this point.

**Default**

If `neighbor keepalives-always` is not specified, it is the same as if the user had specified the following:

```plaintext
(config-router-bgp)# no neighbor [ip_address | group-name] keepalives-always
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example enables the keepalives-always feature for peer 1.2.3.4.

```plaintext
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 keepalives-always
(config-router-bgp)# exit
(config)#
```

**neighbor local-as**

**Name**

`neighbor local-as` - identifies the autonomous system (AS) that Advanced Routing Suite is representing to a peer

**Syntax**

```plaintext
neighbor ip_address local-as as_num
no neighbor ip_address local-as as_num
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `as_num` - a number between 1 and 65535, specifying a set of routers under a single technical administration and assigned by the Internet Assigned Numbers Authority

**Description**

The `neighbor local-as` command identifies the AS that Advanced Routing Suite is representing to a group of peers.

**Note:** This command is valid only for external peers.
**Default**

The default AS number for this command is the current AS (configured with the `router bgp` command in Global Configuration mode).

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example causes BGP to represent itself to the peer 1.2.3.4 as being in AS 100.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 local-as 100
(config-router-bgp)# exit
(config)#
```

**Example 2**

The following example causes BGP to cease representing itself to peer 1.2.3.4 as being in AS 100. Instead, it will represent itself as being in AS 64512.

```
(config)# router bgp 64512
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# no neighbor 1.2.3.4 local-as 100
(config-router-bgp)# exit
(config)#
```

**neighbor log-up-down**

**Name**

`neighbor log-up-down` - causes a message to be logged via the syslog mechanism whenever a BGP peer or peer group enters or leaves the Established state.

**Syntax**

```
neighbor [ ip_address | group-name ] log-up-down
no neighbor [ ip_address | group-name ] log-up-down
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**

`neighbor log-up-down` specifies whether a message will be logged via the syslog mechanism whenever a BGP peer or peer group enters or leaves the Established state.
Default

If neighbor log-up-down is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] log-up-down
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example causes a message to be logged in syslog whenever this peer leaves or enters the Established state.

```
(config)# bgp router 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 log-up-down
(config-router-bgp)# exit
(config)#
```

neighbor maximum-routes

**Name**

neighbor maximum-routes - specifies the maximum number of routes that BGP will accept for installation into the Advanced Routing Suite routing information base (RIB)

**Syntax**

```
neighbor [ip_address | group-name] maximum-routes num
[threshold]? [(restart time-minutes) | (warning-only)]?
```

```
no neighbor [ip_address | group-name] maximum-routes num?
[threshold]? [(restart time-minutes) | (warning-only)]?
```

**Mode**

BGP Router Configuration

**Parameters**

- ip_address - the address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- group-name - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- num - an integer between 1 and 4,294,967,295, inclusive
- threshold - a percentage
- restart time-minutes - an integer from 1 to 65535, inclusive, specifying a number of minutes. This specifies the amount of time that will elapse before a restart occurs.
- warning-only - specifies to log a warning message, stop accepting routes, and keep the established peering session

**Description**

The neighbor maximum-routes command configures the maximum number of routes that a non-group peer or peers in a peer group will accept for installation into Advanced Routing Suite. You can also specify a
threshold percentage. When this percentage of maximum routes are accepted, the configured action will occur. In addition, an action can be configured to occur when this maximum number is exceeded. The action can be one of the following:

- If **warning-only** is specified, log a warning message and stop accepting additional routes, but still keep the established peer session
- If **restart** is specified, close the peer session and restart after the specified number of minutes
- If neither **warning-only** nor **restart** is configured, close the peer session and keep this in idle state until the "clear ip bgp" ("clear ip bgp" on page 94) command is issued

The negative form of this command, no neighbor maximum-routes, removes the configure route limit and returns this setting to its default value of unlimited

**Default**

If **neighbor maximum-routes** is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] max
imum-routes 4294967295
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example causes Advanced Routing Suite to install a maximum of 1000 routes learned from peer 1.2.3.4 into the routing table. If this amount is exceeded, then the BGP peering session will restart after 10 minutes.

```
(config)# router bgp 65535
(config-router-bgp)# bdp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 maximum-routes 1000 restart 10
(config-router-bgp)# exit
(config)#
```

**Example 2**
The following example causes Advanced Routing Suite to install a maximum of 1000 routes learned from peer 10.23.12.13 into the routing table. If more than 80% of these routes are installed, then Advanced Routing Suite will close the peering session and keep it down until a clear ip bgp command is specified.

```
(config)# router bgp 65535
(config-router-bgp)# neighbor 10.23.12.13 remote-as 5
(config-router-bgp)# neighbor 10.23.12.13 maximum-routes 1000 80
(config-router-bgp)# exit
(config)#
```

**Example 3**
The following example causes Advanced Routing Suite to install a maximum of 1000 routes learned from peer 4.3.2.1 into the routing table. If this number is exceeded, then Advanced Routing Suite will log a warning message and stop accepting additional routes.

```
(config)# router bgp 65535
(config-router-bgp)# bdp router-id 1.1.1.1
(config-router-bgp)# neighbor 4.3.2.1 remote-as 5
```
Border Gateway Protocol (BGP)

```
(config-router-bgp)# neighbor 4.3.2.1 maximum-routes 1000 warning-only
(config-router-bgp)# exit
(config)#
```

**neighbor metric-out**

**Name**
neighbor metric-out - causes BGP to send a metric value when routes are advertised to peers

**Syntax**
```
neighbor ip_address metric-out num
no neighbor ip_address metric-out num?
```

**Mode**
BGP Router Configuration

**Parameters**
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `num` - an integer between 0 and 4,294,967,295, inclusive

**Description**
The `neighbor metric-out` command causes a BGP metric (MED) value to be set on routes when they are advertised to peers. The negative form of this command, `no neighbor metric-out`, removes the configured `num` value. **Note:** Specifying a value for `num` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If `neighbor metric-out` is not specified, it is the same as if the user had specified the following:
```
(config-router-bgp)# no neighbor [ip_address | group-name] metric-out
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures a metric value of 50 for routes advertised to peers.
```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 metric-out 50
(config-router-bgp)# exit
(config)#
```

**neighbor multi-protocol-nexthop**

**Name**
neighbor multi-protocol-nexthop - specifies whether to send a standard BGP nexthop when only sending multi-protocol BGP routes
Syntax

neighbor [ip_address | group-name] multi-protocol-nexthop
no neighbor [ip_address | group-name] multi-protocol-nexthop

Mode

BGP Router Configuration

Parameters

ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description

According to the BGP multi-protocol specification RFC 2858, when a BGP Speaker is only sending multi-protocol routes and is not sending any reachability in the standard BGP NLRI field, it is not necessary to send the standard BGP NEXT_HOP path attribute. Unfortunately, some well-deployed implementations have a bug where this field must be present even if it is ignored.

By default, Advanced Routing Suite will send the standard NEXT_HOP field, even when only multi-protocol routes are being sent. If the reachability is IPv4, the NEXT_HOP field will be the same as the multi-protocol nexthop. In the case of non-IPv4 reachability, the standard NEXT_HOP field will contain “0.0.0.0”.

Default

If neighbor multi-protocol-nexthop is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] multi-protocol-nexthop
```

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, a standard BGP nexthop will be sent when only sending multi-protocol BGP routes.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 multi-protocol-nexthop
(config-router-bgp)# exit
(config)#
```

neighbor next-hop-self

Name

neighbor next-hop-self - specifies whether this neighbor’s next hop should be the router’s own address on advertisement

Syntax

```
neighbor [ip_address | group-name] next-hop-self
no neighbor [ip_address | group-name] next-hop-self
```
Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
Use the neighbor next-hop-self command to set this group's next hop to the router's own address even if it would normally be possible to send a third-party next hop. Specifying this command can cause inefficient routes to be followed. It might be needed in some cases to deal with improperly bridged interconnect media (in cases where the routers on the "shared" medium do not really have full connectivity to each other), or when political situations cause broken links.

Default
If neighbor next-hop-self is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] next-hop-self
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that the neighbor will the router's own address as the next hop.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 10.132.10.15
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 next-hop-self
(config-router-bgp)# exit
(config)#
```

neighbor orf comm-list

Name
neighbor orf comm-list - sends a communities ORF message with the communities filter to the specified BGP peer

Syntax
```
neighbor ip_address orf comm-list comm-list-name
no neighbor ip_address orf comm-list comm-list-name
```

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
comm-list-name - the name of a community list
Description
The `neighbor orf comm-list` commands sends a communities ORF message with the communities filter specified by the community list name to the specified peer. The negative form of this command, `no neighbor orf comm-list`, stops the sending of this information.

Note: When configured in BGP Router Configuration mode, the address family sent will be IPv4 unicast.

Default
ORF messages are not sent by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that BGP peer 4.3.2.1 will send and receive ORF capability for community filtering information. In addition, a communities ORF message with the filter specified by community list comm1 will be sent to peer 4.3.2.1.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf comm-filter both
(config-router-bgp)# neighbor 4.3.2.1 orf comm-list comm1
```

neighbor orf extcomm-list

Name
neighbor orf extcomm-list - sends an extended communities ORF message with the filter specified by the extended community list name to the specified BGP peer

Syntax
```
neighbor ip_address orf extcomm-list extcomm-list-name
no neighbor ip_address orf extcomm-list extcomm-list-name
```

Mode
BGP Router Configuration

Parameters
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `extcomm-list-name` - the name of an extended community list

Description
The `neighbor orf extcomm-list` commands sends an extended communities ORF message with the filter specified by the extended community list name to the specified peer. The negative form of this command, `no neighbor orf extcomm-list`, stops the sending of this information.

Note: When configured in BGP Router Configuration mode, the address family sent will be IPv4 unicast.

Default
ORF messages are not sent by default.
**neighor orf prefix-list**

**Name**
`neighbor orf prefix-list` - sends a prefixes ORF message with the filter specified by the prefix list name to the specified BGP peer.

**Syntax**
```
neighbor ip_address orf prefix-list prefix-list-name
no neighbor ip_address orf prefix-list prefix-list-name
```

**Mode**
BGP Router Configuration

**Parameters**
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `prefix-list-name` - the name of a prefix list

**Description**
The `neighbor orf prefix-list` commands sends a prefixes ORF message with the filter specified by the prefix list name to the specified peer. The negative form of this command, `no neighbor orf prefix-list`, stops the sending of this information.

**Note:** When configured in BGP Router Configuration mode, the address family sent will be IPv4 unicast.

**Default**
ORF messages are not sent by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example specifies that BGP peer 4.3.2.1 will send and receive ORF capability for prefixes filtering information. In addition, a prefixes ORF message with the filter specified by prefix list `preflist1` will be sent to peer 4.3.2.1.
```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf extcomm-filter both
(config-router-bgp)# neighbor 4.3.2.1 orf extcomm-list extcomm1
```

The following example specifies that BGP peer 4.3.2.1 will send and receive ORF capability for extended community filtering information. In addition, an extended communities ORF message with the filter specified by extended community list `extcomm1` will be sent to peer 4.3.2.1.
```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 4.3.2.1 remote-as 65001
(config-router-bgp)# neighbor 4.3.2.1 capability orf extcomm-filter both
(config-router-bgp)# neighbor 4.3.2.1 orf extcomm-list extcomm1
```
neighbor out-delay

Name
neighbor out-delay - specifies the amount of time that a route must be present in the Advanced Routing Suite routing database before it is exported to BGP

Syntax
neighbor ip_address [out-delay time-seconds ]
no neighbor ip_address [out-delay time-seconds ]?

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
out-delay time-seconds - the number of seconds, specified as an integer between 0 and 4,294,967,295, inclusive. This option defaults to 0 and is, thus, displayed above as optional.

Description
Use the neighbor out-delay command to damp route fluctuations. The out-delay time is the amount of time that a route must be present in the Advanced Routing Suite routing database before it is exported to BGP.

Note: Weighted Route Damping may be better suited for improving overall network stability. The use of this option may delay route convergence for well-behaved routers.

Default
This command is not explicitly configured by default; however, if a neighbor address is configured, the out-delay time will default to 0.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example enables an out-delay value of 10 seconds on the neighbor 1.2.3.4

(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 out-delay 10
(config-router-bgp)# exit
(config)#
neighbor passive

Name
neighbor passive - prevents Advanced Routing Suite from ever trying to open a BGP connection with peers in this group

Syntax

    neighbor [ ip_address | group-name ] passive
    no neighbor [ ip_address | group-name ] passive

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
The neighbor passive command prevents Advanced Routing Suite from trying to initiate a BGP connection with peers in this group. Instead, the router will wait for the peer to initiate a connection.

This command was introduced to handle a problem in BGP3 and earlier, in which two peers might both attempt to initiate a connection at the same time. This problem has been correct in the BGP4 protocol, and, thus, this command is not needed with BGP 4 sessions.

Note: If the neighbor passive command is applied to both sides of a peering session, the session will never be established. For this reason, and because it is generally not needed, the use of neighbor passive is discouraged.

Default
If neighbor passive is not specified, it is the same as if the user had specified the following:

    (config-router-bgp)# no neighbor [ip_address | group-name] passive

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, BGP will never initiate a connection with the peer.

    (config)# router bgp 65535
    (config-router-bgp)# bgp router-id 1.1.1.1
    (config-router-bgp)# neighbor 1.2.3.4 remote-as 5
    (config-router-bgp)# neighbor 1.2.3.4 passive
    (config-router-bgp)# exit
    (config)#

neighbor password

Name
neighbor password - configures a TCP-MD5 password for a peer or peer group
Syntax

```
neighbor [ ip_address ] password key
no neighbor [ ip_address ] password key
```

Mode
BGP Router Configuration

Parameters
- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `key` - a string of characters

Description
The `neighbor password` command allows you to set a TCP-MD5 password for a peer or peer group. This feature requires OS support of the TCP MD5 Signature Extension (RFC 2385). **Note:** You must first configure either a remote AS or a peer group before configuring an authentication password.

Default
A password is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples

**Example 1**
The following example configures a TCP-MD5 password of "abc" for this peer.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 password abc
(config-router-bgp)# exit
(config)#
```

**Example 2**
The following example removes the authentication for peer 1.2.3.4.

```
(config)# router bgp 65535
(config-router-bgp)# no neighbor 1.2.3.4 password abc
(config-router-bgp)# exit
(config)#
```

**neighbor pedantic**

Name
- `neighbor pedantic` - turns on extra, non-critical logging about malformed BGP update messages that are being ignored

Syntax

```
neighbor [ ip_address | group-name ] pedantic
```
no neighbor [ip_address | group-name] pedantic

**Mode**

BGP Router Configuration

**Parameters**

*ip_address* - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

*group-name* - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**

This command turns on extra non-critical logging about malformed BGP updates that are being ignored.

**Default**

If `neighbor pedantic` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] pedantic
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example turns on pedantic logging for this neighbor.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 pedantic
(config-router-bgp)# exit
```

**neighbor peer-group**

**Name**

`neighbor peer-group` - creates a BGP peer group

**Syntax**

```
neighbor group-name peer-group
no neighbor group-name peer-group

neighbor ip_address peer-group pgname
no neighbor ip_address peer-group pgname
```

**Mode**

BGP Router Configuration

**Parameters**

*group-name* - a string of characters indicating the name of a BGP peer group

*ip_address* - an IPv4 address to add to a peer group. IPv4 addresses are specified in dotted-quad format.
pgname - the name of a peer group into which you want the neighbor added. The peer group name is specified using the `neighbor name peer-group` command, and that command must be specified before neighbors can be added to it.

**Description**

Use the `neighbor peer-group` command to create a peer group. BGP peers can then be added to that group. After this command is issued, peers are added to the group using the `neighbor ip_address peer-group name` command. All peers within a group must have the same AS number.

**Note:** Once you put a neighbor in a group, it loses all options previously configured on the neighbor, except for its AS number. (See Example 2) Thus, changes to neighbor configuration options should be entered after putting the neighbor in a group.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures peer group "abc". Peers 1.2.3.4 and 4.3.2.1 are then added to this peer group.

```
(config)# router bgp 7345
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 4.3.2.1 remote-as 5
(config-router-bgp)# neighbor abc peer-group
(config-router-bgp)# neighbor 1.2.3.4 peer-group abc
(config-router-bgp)# neighbor 4.3.2.1 peer-group abc
(config-router-bgp)# exit
(config)#
```

**Example 2**

The following example configures a peer group called xyz. Peer 10.133.10.23 is configured separately, then a route map named "import" is imported into that peer. The peer is then added to the peer group. The result of adding this neighbor to peer group xyz is that the route-map setting will be lost. This is because once you put a neighbor in a group, it loses all options previously configured on the neighbor except for its AS number.

```
(config)# router bgp 201
(config-router-bgp)# neighbor xyz peer-group
(config-router-bgp)# neighbor 10.133.10.23 remote-as 201
(config-router-bgp)# neighbor 10.133.10.23 route-map import in
(config-router-bgp)# neighbor 10.133.10.23 peer-group xyz
(config-router-bgp)# exit
(config)#
```
**neighbor preference2**

**Name**

`neighbor preference2` - breaks a preference tie between two groups in a BGP peer

**Syntax**

```
neighbor [ip_address | group-name] preference2 pref2
no neighbor [ip_address | group-name] preference2 pref2
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `pref2` - an integer between 0 and 255, inclusive

**Description**

The `neighbor preference2` command breaks a preference tie between groups in a BGP peer. Preferences are the first criteria of comparison for route selection. When a route has been learned from more than one protocol, the active route will be selected from the protocol with the lowest preference.

The negative form of this command, `no preference2`, removes the configured value and returns this to its default value of 0.

**Default**

If `neighbor preference2` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# neighbor [ip_address | group-name] preference2 0
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the global BGP preference2 value to be 20 and the BGP preference2 value for peer 1.2.3.4 to be 10.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# preference2 20
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 preference2 10
(config-router-bgp)# exit
(config)#
```
neighbor receive-buffer

Name
neighbor receive-buffer - controls the amount of memory requested from the kernel for the receive buffer

Syntax

```
neighbor [ip_address | group-name] receive-buffer kbufsize
no neighbor [ip_address | group-name] receive-buffer kbufsize
```

Mode
BGP Router Configuration

Parameters

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `group-name` - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- `kbufsize` - kernel send-buffer size in bytes, specified as an integer between 4096 and 4,294,967,295, inclusive

Description

The `neighbor receive-buffer` command controls the amount of memory requested from the kernel for the receive buffer. The maximum supported is 4,294,967,295 bytes; although many kernels have a lower limit. By default, Advanced Routing Suite configures the maximum supported. This command is not necessary on normally functioning systems.

Default

By default, Advanced Routing Suite configures the maximum supported. The maximum supported is 4,294,967,295 bytes; although many kernels have a lower limit.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example limits the receive buffer to 5000 bytes for this peer.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 receive-buffer 5000
(config-router-bgp)# exit
(config)#
```

neighbor remote-as

Name

neighbor remote-as - configures the remote AS for an IPv4 peer
Syntax

neighbor ip_address remote-as as_num
no neighbor ip_address remote-as as_num

Mode
BGP Router Configuration

Parameters

ip_address - a valid IPv4 address specified in dotted-quad format
remote-as as_num - the autonomous system (AS) number of a BGP peer. This can be an integer from 1 to 65535, inclusive.

Description

The neighbor remote-as command configures the IPv4 address of the host machine. You must specify this command before configuring any other BGP peer commands.

Default

BGP peers are not explicitly configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example configures a BGP peer with an IPv4 address of 1.2.3.4 and an AS number of 5.

    (config)# router bgp 64512
    (config-router-bgp)# bgp router-id 1.1.1.1
    (config-router-bgp)# neighbor 1.2.3.4 remote-as 5
    (config-router-bgp)# exit
    (config)#

neighbor remove-private-as

Name

neighbor remove-private-as - specifies whether to remove private autonomous system (AS) numbers from outbound updates to a peer or peer group

Syntax

neighbor [ip_address | group-name] remove-private-as
no neighbor [ip_address | group-name] remove-private-as

Mode

BGP Router Configuration

Parameters

ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**

Use the `neighbor remove-private-as` command to remove private AS numbers when sending updates to a peer or peer group.

**Default**

If `neighbor remove-private-as` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor [ip_address | group-name] remove-private-as
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example causes BGP to strip private AS numbers from updates to this peer.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 remove-private-as
(config-router-bgp)# exit
(config)#
```

**Example 2**

The following example causes BGP to no longer remove private AS numbers.

```
(config)# router bgp 64512
(config-router-bgp)# no neighbor 1.2.3.4 remove-private-as
(config-router-bgp)# exit
(config)#
```

**neighbor route-map**

**Name**

`neighbor route-map` - specifies a route map for filtering routes to or from a BGP peer

**Syntax**

```
neighbor ip_address route-map rm-name [in | out]
no neighbor ip_address route-map rm-name [in | out]
```

**Mode**

BGP Router Configuration

**Parameters**

- `ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- `rm-name` - the name of a configured route map
in | out - specify whether the route map should be applied to routes being learned from (in) or sent to (out) BGP

Description
Route maps are used to control the redistribution of routes between protocols. After configuring a route map, it can then be specified in BGP. Use the neighbor route-map command to specify a configured route map to be exported into or out of BGP.

Default
Route Maps are not explicitly specified by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, a configured route map named “abc” is specified to be exported into BGP.

```
(config)# router bgp 4444
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 route-map abc in
(config-router-bgp)# exit
```

neighbor route-reflector-client

Name
neighbor route-reflector-client - specifies that Advanced Routing Suite will act as a route reflector for this neighbor

Syntax
```
neighbor [ ip_address | group-name ] route-reflector-client [meshed]?
no neighbor [ ip_address | group-name ] route-reflector-client [meshed]?
```

Mode
BGP Router Configuration
Address Family Configuration

Parameters
- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- **group-name** - the name of a BGP peer group specified as a string of characters. **Note**: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- **meshed** - optionally specify whether the reflector client is meshed

Description
The neighbor route-reflector-client command specifies that Advanced Routing Suite will act as a route reflector for this group. Configuring the negative of this command specifies that Advanced Routing Suite will not act as an intra-group reflector and thus will not reflect routes back to peers within the same group. This is used when client peers within a route-reflection group are fully meshed.
Default
Route reflector clients are not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that the neighbor 1.2.3.4 is to be a route reflector client.

```
(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 route-reflector-client
(config-router-bgp)# exit
(config)#
```

neighbor route-to-peer

**Name**
neighbor route-to-peer - specifies the actual time to live (TTL) used on a socket in all cases

**Syntax**
```
neighbor [ ip_address | group-name ] route-to-peer
no neighbor [ ip_address | group-name ]route-to-peer
```

**Mode**
BGP Router Configuration

**Parameters**
- **ip_address** - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- **group-name** - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

**Description**
The `neighbor route-to-peer` command specifies the actual TTL used on a socket in all cases. In particular, if Advanced Routing Suite realizes that two BGP speakers are peering over a single network, Advanced Routing Suite automatically sets the `dontroute` option on their peering session. This, in turn, causes the TTL of the packets to be set to 1. The `neighbor route-to-peer` command prevents the `dontroute` option from being set. If you specify `route-to-peer`, but do not specify a TTL, and you are directly connected, then Advanced Routing Suite will set the TTL of your peering session to 1. If you want a TTL greater than 1 for directly connected peers, you must specify both `route-to-peer` and the `ttl_num` that you require. Refer to the “neighbor ttl” ("neighbor ttl" on page 142) command.

**Default**
If `neighbor route-to-peer` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor ip_address route-to-peer
```

**Command History**
NGC 2.2 - This command was introduced.
Examples
The following example enables the route-to-peer option for this peer.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 route-to-peer
(config-router-bgp)# exit
(config)#
```

neighbor send-buffer

**Name**
neighbor send-buffer - controls the amount of send buffering asked of the kernel

**Syntax**
```
neighbor [ip_address | group-name] send-buffer kbufsize
no neighbor [ip_address | group-name] send-buffer kbufsize
```

**Mode**
BGP Router Configuration

**Parameters**
- ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
- group-name - the name of a BGP peer group specified as a string of characters. **Note:** If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
- kbufsize - kernel send-buffer size in bytes, specified as an integer between 4096 and 4,294,967,295, inclusive

**Description**
The `neighbor send-buffer` command controls the amount of memory requested from the kernel for the send buffer. The maximum supported is 4,294,967,295 bytes; although many kernels have a lower limit. By default, Advanced Routing Suite configures the maximum supported. This command is not needed on normally functioning systems.

**Default**
By default, Advanced Routing Suite configures the maximum supported. The maximum supported is 4,294,967,295 bytes; although many kernels have a lower limit.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example limits the send buffer to 5000 bytes for this peer.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 send-buffer 5000
```
neighbor send-community

**Name**
neighbor send-community - specifies whether communities are permitted to be sent

**Syntax**

```
neighbor ip_address send-community
no neighbor ip_address send-community
```

**Mode**
BGP Router Configuration

**Parameters**

`ip_address` - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

**Description**
Use `neighbor send-community` to specify whether communities are permitted in a peer.

**Default**
If `neighbor send-community` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# no neighbor ip_address send-community
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example causes Advanced Routing Suite to send any specific community values to peer 1.2.3.4.

```
(config)# router bgp 60001
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 send-community
(config-router-bgp)# exit
(config)#
```

**Example 2**
The following example causes Advanced Routing Suite to not send any communities to this peer.

```
(config)# router bgp 60001
(config-router-bgp)# no neighbor 1.2.3.4 send-community
(config-router-bgp)# exit
(config)#
```
neighbor soft-reconfiguration inbound

Name
neighbor soft-reconfiguration inbound - activates the route refresh capability for a peer or peer group

Syntax
neighbor [ip_address | group-name] soft-reconfiguration inbound
no neighbor [ip_address | group-name] soft-reconfiguration inbound

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
Use the neighbor soft-reconfiguration inbound command to activate the route refresh capability for this peer.

Default
If neighbor soft-reconfiguration inbound is not specified, it is the same as if the user had specified the following:

    (config-router-bgp)# no neighbor [ip_address | group-name] soft-reconfiguration inbound

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns on the route refresh capability for peer 1.2.3.4.

    (config)# router bgp 60101
    (config-router-bgp)# bgp router-id 1.1.1.1
    (config-router-bgp)# neighbor 1.2.3.4 remote-as 5
    (config-router-bgp)# neighbor 1.2.3.4 soft-reconfiguration inbound
    (config-router-bgp)# exit
    (config)#

neighbor timers

Name
neighbor timers - specifies holdtime and keepalive time values within a BGP peer

Syntax
neighbor [ip_address | group-name] timers
[ keepalive_value holdtime_value ]
no neighbor [ip_address | group-name] timers
[ keepalive_value holdtime_value ]?

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
keepalive_value - the time, in seconds, between keepalive messages with a value range between 0 and 4,294,967,295, inclusive
holdtime_value - the time, in seconds of the BGP Hold Timer with a value of 0 or a value between 3 and 65535, inclusive

Description
The BGP keepalive timer will be set to one-third of the negotiated holdtime by default. Use keepalive to specify the number of seconds that will elapse between keepalive messages.

The holdtime value specifies the number of seconds to use when negotiating a peering session within this group. If Advanced Routing Suite does not receive a keepalive, update, or notification message within the specified period, then the BGP connection will be closed.

The negotiated holdtime value is the lesser of the values sent in the exchanged BGP Open messages. If a time of zero is specified, no keepalives will be sent. If a time of zero is received from the remote peer, then the holdtime must be configured to be zero in order for the peering session to become established.

Note: You cannot specify a holdtime value of 1 or 2. Attempting to do so will result in an error.

The negative form of this command, no neighbor timers, removes the configured values and returns this to its default values of 180 and 60 seconds, respectively. Note: Specifying a value for holdtime and keepalive in the no form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default
If neighbor timers is not specified, it is the same as if the user had specified the following:
(config-router-bgp)# neighbor [ip_address | group-name] timers 180 60

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a keepalive value of 40 seconds and a holdtime value of 90 seconds for this peer.

(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 timers 40 90
(config-router-bgp)# exit
(config)#
neighbor ttl

Name
neighbor ttl - specifies time to live (TTL) value

Syntax
neighbor [ip_address | group-name] ttl ttl_num
no neighbor [ip_address | group-name] ttl ttl_num

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
ttl_num - an integer from 1 to 255 specifying either a number of seconds or a number of hops

Description
The neighbor ttl command is provided mainly for attempting to communicate with improperly functioning routers that ignore packets sent with a TTL value of 1. The ttl_num value has two units: seconds and number of hops. Either of these can be used. Note: Not all kernels allow the TTL to be specified for TCP connections.

Default
By default, Advanced Routing Suite sets the IP TTL for local peers to 1 and the TTL for non-local peers to the default kernel value.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a TTL value of 5 for this peer.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 ttl 5
(config-router-bgp)# exit
(config)#
```
no neighbor [ip_address | group-name] update-source source-addr

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format

Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

source-addr - the local IP interface address

group-name - the name of a BGP peer group specified as a string of characters.

Description
Use neighbor update-source to specify the IP address to be used on the local end of the TCP connection with the peer. This is the address of a broadcast, NBMA or loopback interface and the local address of a point-to-point interface.

For external peers, the local address must be on an interface that is shared with the peer or with the peer’s gateway when a gateway is used. A session with an external peer will be opened only when an interface with the appropriate local address (through which the peer or gateway address is directly reachable) is operating. For internal peers, a peer session will be maintained when any interface with the specified local address is operating. In any case, an incoming connection will be recognized as a match for a configured peer only if it is addressed to the configured local address.

Default
The default update-source address is the IP address of a shared interface.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example causes the TCP session to peer 1.2.3.4 to be established over the interface 4.3.2.1.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.1.1.1
(config-router-bgp)# neighbor 1.2.3.4 remote-as 5
(config-router-bgp)# neighbor 1.2.3.4 update-source 4.3.2.1
(config-router-bgp)# exit
(config)#
```

Example 2
The following example causes the TCP session to be established over the shared interface.

```
(config)# router bgp 65535
(config-router-bgp)# no neighbor 1.2.3.4 update-source 4.3.2.1
(config-router-bgp)# exit
(config)#
```
neighbor use-med

Name
neighbor use-med - specifies whether MEDs will be used in routing computations

Syntax
neighbor [ip_address | group-name] use-med
no neighbor [ip_address | group-name] use-med

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.

Description
By default, any MED (Multi Exit Disc) received on a BGP connection is ignored. If MEDs are to be used in selecting routes, the use-med option must be specified. When two routes to the same destination are received from different peers within the same peer-as, they could have different MEDs. When choosing between these routes, assuming that nothing else makes one preferable to the other (such as configured policy), the values of the differing MEDs are used to choose which route to use. In this comparison, the route with the lowest MED is preferred. Routes without MEDs are treated as having a MED value of zero unless "bgp bestpath med missing-as-worst" ("bgp bestpath med missing-as-worst" on page 83) is configured on.

Default
If neighbor use-med is not specified, it is the same as if the user had specified the following:
   (config-router-bgp)# no neighbor [ip_address | group-name] use-med

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example causes BGP to pay attention to MED values received from this peer.
   (config)# router bgp 65535
   (config-router-bgp)# bgp router-id 1.1.1.1
   (config-router-bgp)# neighbor 1.2.3.4 remote-as 5
   (config-router-bgp)# neighbor 1.2.3.4 use-med
   (config-router-bgp)# exit
   (config)#

Example 2
The following example causes BGP to ignore MED values received from this peer.
   (config)# router bgp 65535
   (config-router-bgp)# no neighbor 1.2.3.4 use-med
neighbor v4-gateway

Name
neighbor v4-gateway - configures an IPv4 intermediate destination by which packets are delivered to
their ultimate destination

Syntax
neighbor [ip_address | group-name] v4-gateway gwv4_address
no neighbor [ip_address | group-name] v4-gateway gwv4_address

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP
address is specified (for example 256.1.1.1), it will be read as a group name.
gwv4_address - the IPv4 address of the gateway specified in dotted-quad format

Description
A gateway is an intermediate destination by which packets are delivered to their ultimate destination. This
command instructs Advanced Routing Suite to use a form of multi-hop EBGP.

If a network is not shared with this group, then this command specifies a router on an attached network to
be used as the next hop router for routes received from this peer.

The gateway can also be used to specify a next hop for groups that are on shared networks. For example,
you might use gateway to ensure that third-party next hops are never accepted from a given group by
specifying that group’s address as its own gateway. The gateway specified must have consistent routing
information to prevent routing loops. In most cases, the gateway is not needed.

Default
The neighbor v4-gateway command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures address 192.168.1.1 as a gateway address for peer 1.1.1.1.

(config)# router bgp 64512
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# neighbor 1.1.1.1 remote-as 5
(config-router-bgp)# neighbor 1.1.1.1 v4-gateway 192.168.1.1
(config-router-bgp)# exit
(config)#
neighbor version

Name
neighbor version - specifies the BGP version number to use

Syntax
neighbor [ip_address | group-name] version num

Mode
BGP Router Configuration

Parameters
ip_address - the IP address of a BGP peer specified as a valid IPv4 address in dotted-quad format
group-name - the name of a BGP peer group specified as a string of characters. Note: If an invalid IP address is specified (for example 256.1.1.1), it will be read as a group name.
num - the BGP version number. Currently, this can only be 4.

Description
Use neighbor version to specify the version of BGP you want to run on a peer group or peer. Because only version 4 is currently supported, there is no negative form of this command available.

Default
If neighbor version is not specified, it is the same as if the user had specified the following:

   (config-router-bgp)# neighbor [ip_address | group-name] version 4

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures BGP version 4 on peer 1.2.3.4.

   (config)# router bgp 65534
   (config-router-bgp)# bgp router-id 1.1.1.1
   (config-router-bgp)# neighbor 1.2.3.4 remote-as 5
   (config-router-bgp)# neighbor 1.2.3.4 version 4
   (config-router-bgp)# exit

network

Name
network - specifies prefixes to be imported into BGP

Syntax
network (ipv4_address (mask net-mask))[route-map name]? [restrict]?
   no network (ipv4_address (mask net-mask))[route-map name]? [restrict]?

Mode
BGP Router Configuration
Address Family Mode

Parameters

ipv4_address (mask net-mask) - specify an IPv4 address with an optional mask value. The ipv4_address and net-mask values are specified in dotted-quad format

route-map name - optionally specify a route map to apply policy to the prefix

restrict - optionally specify that the matching network range should not be advertised to other peers

Description

The network command is used to specify prefixes that should be imported into BGP. The negative form of this command removes configured network prefixes.

Note: Some competitors set the BGP ORIGIN attribute differently on routes configured via the static statement (origin incomplete) than via the BGP network statement (origin IGP). Advanced Routing Suite always sets origin incomplete. Use the "set origin" Route Map command to modify the origin.

Default

Network prefixes are not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example imports the network 10.1.0.0 with a mask of 255.255.255.0 into BGP. Additionally, this network range will be advertised to other peers.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# network 10.1.0.0 mask 255.255.255.0 permit
(config-router-bgp)# exit
(config)#
```

See Also

Chapter 31 Route Maps

preference2

Name

preference2 - breaks a preference tie between two groups

Syntax

```
preference2 pref2
no preference2 pref2?
```

Mode

BGP Router Configuration

Parameters

pref2 - an integer between 0 and 255, inclusive
Description

The `preference2` command breaks a preference tie between groups. Preferences are the first criteria of comparison for route selection. When a route has been learned from more than one group, the active route will be selected from the group with the lowest preference.

The negative form of this command, `no preference2`, removes the configured value and returns this to its default value of 0. **Note:** Specifying a value for `pref2` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `preference2` is not specified, it is the same as if the user had specified the following:

```
(config-router-bgp)# preference2 0
```

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example configures the global BGP preference2 value to be 20.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# preference2 20
(config-router-bgp)# exit
(config)#
```

Example 2

The following example removes the configured BGP preference2 value and reverts it back to its default value.

```
(config)# router bgp 65535
(config-router-bgp)# no preference2 20
(config-router-bgp)# exit
(config)#
```

redistribute

Name

`redistribute` - specifies routes to export to the current protocol

Syntax

```
redistribute protocol [route-map name]? (0,10)
no redistribute protocol [route-map name]?
```

Mode

BGP Router Configuration

Parameters

`protocol` - the protocol name whose routes you want to redistribute to the current protocol being configured. Valid protocols are aggregate, bgp, isis, kernel, ospf, ospf-ase, rip, ripng, and static.
route-map name - the name of a route map to apply to these routes. Specifying this is optional.

{0,10} - although this command can be given multiple times, it can only be given once for each of the configurable protocols. In other words, if a redistribute command is given for a protocol and route map, and then given again for the same protocol with a different route map, the second configuration overrides the first.

Description

Use the redistribute command to specify routes to export into the BGP. This command causes routes from the specified protocol to be considered for redistribution into BGP. Additionally, if a route map is specified, then routes from the specified protocol that match the named route map will be considered for redistribution into the current protocol. If the referenced route map has not yet been configured, then an empty route map is created with the specified name.

Note: Configuring this away from its default removes the implicitly configured default. You will have to go back and specify to redistribute BGP routes after the first redistribute configuration in order to export those routes.

Default

The default is to export BGP routes. Note that this is an implicit default that is wiped away with the first redistribute configuration.

Example

Example 1

In the following example BGP is configured to redistribute all RIP and RIPng routes.

```
(config)# router bgp 65535
(config-router-bgp)# redistribute rip
(config-router-bgp)# redistribute ripng
(config-router-bgp)# exit
(config)#
```

Example 2

In the following example, the route map set med is applied to OSPF routes and exported into BGP.

```
(config)# route map setmed
(config-route-map)# set med 20
(config-route-map)# exit
(config)# router bgp
(config-router-bgp)# redistribute ospf route-map setmed
(config-router-bgp)# exit
(config)#
```

Example 3

In the following example, route map "abc" is configured with the following match criteria:

If a route matches interface "fxp1", then communities specified in community "com-set-1" will be added to the route and the metric of the route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface ffp1
(config-route-map)# set community-set com-set-1
(config-route-map)# set metric 50
(config-route-map)# exit
```
This route map is then applied to static routes and exported into BGP.

```
(config)# router bgp 65121
(config-router-bgp)# redistribute kernel route-map abc
(config-router-bgp)# exit
```

Note - Because static routes are installed and identified as kernel routes, use `redistribute kernel` to redistribute static routes.

See Also
"distribute-list" ("distribute-list" on page 97)

**router bgp**

**Name**

`router bgp` - configures BGP on a router and specifies the router’s autonomous system number

**Syntax**

```
router bgp as_number
no router bgp as_number
```

**Mode**

Global Configuration

**Parameters**

`as_number` - an integer between 1 and 65535, inclusive

**Description**

The `router bgp` command enables BGP on a router. Because there is no default `as_number`, an autonomous system number must also be specified. The negative form of this command, `no router bgp`, disables BGP on a router.

**Default**

BGP is disabled by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures the router to be a BGP speaker in AS number 65535.

```
(config)# router bgp 65535
(config-router-bgp)#
```

**Example 2**

The following example turns off BGP on the router.
(config)# no router bgp 65535
(config)#

show ip bgp

Name
show ip bgp - displays information about BGP routes installed in the BGP routing information base (RIB)

Syntax
    show ip bgp [ip_address]? [prefix (longer-prefixes?)]

Mode
User Execution

Parameters
ip_address - optionally specify an IP address
prefix - optionally specify a prefix with an address and a mask length
longer-prefixes - if a prefix is specified, optionally specify to show routes matching the specified Network/Mask pair only

Description
Use the show ip bgp query to obtain information about BGP routes installed in the BGP RIB.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example returns BGP information for all BGP routes installed in the BGP RIB.

> show ip bgp

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>R</th>
<th>Metric</th>
<th>LocPref</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.3.4/24</td>
<td>192.165.251.1</td>
<td>u</td>
<td>9</td>
<td>100</td>
<td>201 202</td>
</tr>
<tr>
<td>1.2.3.5/24</td>
<td>192.165.251.1</td>
<td>u</td>
<td>10</td>
<td>100</td>
<td>201 202</td>
</tr>
</tbody>
</table>

Field Descriptions
The following table describes the fields that appear in the BGP Query.

Table 13-1 BGP Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>The value of a BGP route</td>
</tr>
<tr>
<td>Next Hop</td>
<td>The Next Hop address corresponding to the BGP route</td>
</tr>
<tr>
<td>R</td>
<td>Specifies whether the route is installed in the unicast or multicast RIB</td>
</tr>
</tbody>
</table>
### show ip bgp instance

**Name**
show ip bgp instance - shows current BGP instance information

**Syntax**

```
show ip bgp instance
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use the `show ip bgp instance` query to obtain summarized information about the current BGP instance.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example shows information about the current BGP instance.

```
> show ip bgp instance
Instance ID is 0
```

### show ip bgp neighbors

**Name**
show ip bgp neighbors - displays information about the state of BGP's IPv4 peering sessions

**Syntax**

```
show ip bgp neighbors [ipv4-address]?
```

**Mode**
Privileged Execution

**Parameters**
ipv4-address - optionally enter an IPv4 address to view information for only the specified neighbor
Description
Use the `show ip bgp neighbors` query to obtain detailed information about the state of BGP's peering sessions. You can optionally specify an IPv4 address. When you do so, the output of the query will display information for the specified address.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example shows detailed information about all BGP peering sessions.

```bash
# show ip bgp neighbors
BGP neighbor is 10.11.31.32, remote AS 65534
BGP version is 4, remote router ID 10.133.10.32
Negotiated version is 4
TTL is 0
holdtime is 180
restart-time is 0
Restarting: no
Current state is "Established"
Last state was "OpenConfirm"
Last event was "RecvKeepAlive"
Last error code was 0
Last error subcode was 0
Local address is 10.11.31.31
Local AS is 65535
Local router ID is 1
92.168.11.31
Capabilities:
Multicprotocol IPv4 Unicast: no
Graceful Restart IPv4 Unicast: no
Multiprotocol IPv4 Multicast: no
Graceful Restart IPv4 Multicast: no
Route Refresh: no
Send End-of-RIB messages: no
Dynamic Capabilities: no
BGP neighbor is 10.11.31.33, remote AS 65533
BGP version is 4, remote router ID 192.168.11.33
Negotiated version is 4
TTL is 0
holdtime is 180
restart-time is 0
Restarting: no
Current state is "Established"
Last state was "OpenConfirm"
Last event was "RecvKeepAlive"
Last error code was 0
Last error subcode was 0
Local address is 10.11.31.31
Local AS is 65535
Local router ID is 192.168.11.31
Capabilities:
Multicprotocol IPv4 Unicast: no
Graceful Restart IPv4 Unicast: no
Multiprotocol IPv4 Multicast: no
Graceful Restart IPv4 Multicast: no
Route Refresh: no
Send End-of-RIB messages: no
Dynamic Capabilities: no
```
**show ip bgp orf**

**Name**
show ip bgp orf - displays ORF-related information for all peers or for a specific peer

**Syntax**
```
show ip bgp orf ip_address?
```

**Mode**
User Execution

**Parameters**
ip_address - optionally specify the IPv4 address of a peer for which you want ORF information returned

**Description**
Use the `show ip bgp orf` query to view ORF-related information for a single peer or for all peers.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
In the following example, ORF information for all peers is displayed.
```
> show ip bgp orf
  ORFs for peer 1.2.3.4
  prefix 10.0.0.0/8 ge 8 le 8 deny
  community 200:300
  extcommunity rt 192.168.10.2:500
  ORFs for peer 2.2.2.2
  community 202:302
  community 400:500
```

**Example 2**
In the following example, ORF information is returned for peer 2.2.2.2 only.
```
> show ip bgp orf 2.2.2.2
  ORFs for peer 2.2.2.2
  community 202:302
  community 400:500
```

**Field Descriptions**
The following table describes the fields that appear in the BGP ORF Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORFs for peer</td>
<td>The IPv4 address of the BGP peer</td>
</tr>
</tbody>
</table>
### show ip bgp paths

**Name**

*show ip bgp paths* - displays BGP AS Path information

**Syntax**

```
show ip bgp paths
```

**Mode**

User Execution

**Parameters**

none

**Description**

Use the *show ip bgp paths* query to view AS Path information.

**Command History**

NGC 2.3 - This command was introduced.

**Examples**

The following example is a query for AS Paths.

```
> show ip bgp paths
      Refcount Metric     Path
      5        0          IGP (Id 2)
      12       0          Incomplete (Id 1)
```

### show ip bgp peer-group

**Name**

*show ip bgp peer-group* - displays information about BGP peer groups

**Syntax**

```
show ip bgp peer-group [name]?
```

**Mode**

User Execution

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Prefix filter information, including the prefix, mask, permit or deny, and the range</td>
</tr>
<tr>
<td>community</td>
<td>The community filter</td>
</tr>
<tr>
<td>extcommunity</td>
<td>The extended community filter</td>
</tr>
</tbody>
</table>
**Parameters**

name - optionally specify the name of a peer group

**Description**

Use the `show ip bgp peer-group` query to obtain BGP peer group information. If a `name` is not specified, then information about all BGP peer groups is displayed.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example returns information for all BGP peer groups.

```
> show ip bgp peer-group
BGP peer-group is abc
BGP version 4
  For address family: IPv4 Unicast
BGP neighbor is abc, peer-group external, members:
  192.168.100.1
  192.168.100.2
```

**Field Descriptions**

The following table describes the fields that appear in the BGP Peer Group Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>The version of BGP that this group is using</td>
</tr>
<tr>
<td>peer-group external/internal</td>
<td>The type of this peer group</td>
</tr>
<tr>
<td>members:</td>
<td>List of IP addresses of peers that belong to this group</td>
</tr>
</tbody>
</table>

**show ip bgp summary**

**Name**

`show ip bgp summary` - shows summarized BGP information

**Syntax**

```
show ip bgp summary
```

**Mode**

User Execution

**Parameters**

none
Description
Use the `show ip bgp summary` command to view summarized BGP information, including the router ID and the local AS number.

Command History
NGC 2.2 - This command was introduced

Examples
The following example shows a request for summarized BGP information.

```
> show ip bgp summary
BGP router identifier 65.247.36.97, local AS number 4
>
```

timers bgp

Name
`timers bgp` - specifies global keepalive and holdtime values for BGP

Syntax
```
timers bgp keepalive_value holdtime_value
no timers bgp [keepalive_value holdtime_value]?
```

Mode
BGP Router Configuration

Parameters
- `keepalive_value` - the time, in seconds, between keepalive messages with a value range between 0 and 4,294,967,295, inclusive
- `holdtime_value` - the time, in seconds of the BGP Hold Timer with a value of 0 or a value between 3 and 65535, inclusive

Description
The BGP keepalive timer will be set to one-third of the negotiated holdtime by default. The keepalive value specifies the number of seconds that will elapse between keepalive messages.

The holdtime value specifies the number of seconds to use when negotiating a peering session. If Advanced Routing Suite does not receive a keepalive, update, or notification message within the specified period, then the BGP connection will be closed. The negotiated holdtime value is the lesser of the values sent in the exchanged BGP Open messages. If a time of zero is specified, no keepalives will be sent. If a time of zero is received from the remote peer, then the holdtime must be configured to be zero in order for the peering session to become established.

Note: You cannot specify a holdtime value of 1 or 2. The value can be 0 or at least 3.

The negative form of this command, `no timers bgp`, removes the configured values and returns this to its default values of 180 and 60 seconds, respectively. **Note:** Specifying a value for keepalive and holdtime in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `timers bgp` is not specified, it is the same as if the user had specified the following:
```
(config-router-bgp)# timers bgp 180 60
```
**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example sets the BGP keepalive value to 30 seconds and the holdtime value to 90 seconds. Notice that the values have positional significance. The keepalive value must appear first, and the holdtime value must appear second.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# timers bgp 30 90
(config-router-bgp)# exit
(config)#
```

**Example 2**
The following example removes the user-configured timer values and returns these to their defaults.

```
(config)# router bgp 65535
(config-router-bgp)# no timers bgp
(config-router-bgp)# exit
(config)#
```

**trace file**

**Name**
trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed

**Syntax**
```
trace file file_name [no-timestamp || overwrite]?
no trace file file_name [no-timestamp || overwrite]?
```

**Mode**
BGP Router Configuration

**Parameters**

- file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
- no-timestamp - specifies that a timestamp should not be prepended to all trace lines
- overwrite - specifies to begin tracing by appending or truncating an existing file

**Description**
The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For BGP, the `trace file` command in BGP Router Configuration Mode specifies a file for tracing of all BGP events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `trace flag` command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.
The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file. **Note:** These options are not cumulative across multiple commands. Consider the following example:

```
(config-router-bgp)# trace file /var/log/bgp.log no-timestamp
(config-router-bgp)# trace file /var/log/bgp.log max-files 10
```

The option given in the second command completely replaces that given in the first. In order to specify both `no-timestamp` and `max-files 10`, they must be entered on the same line as follows.

```
(config-router-bgp)# trace file /var/log/bgp.log max-files 10 no-timestamp
```

**Default**

BGP tracing is turned off by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, BGP tracing is written to the file "/var/tmp/bgp.log". No timestamp will display at the beginning of the trace lines.

```
(config)# router bgp 65535
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# trace file /var/tmp/bgp.log no-timestamp
```

---

### trace flag

**Name**

`trace flag` - specifies BGP-specific tracing options as well as options that are common across all protocols.

**Syntax**

```
trace flag ( [ route | normal | state | policy | task | timer | all | aspath ] ) | ( [ packets | open | update | keepalive | notify ] [ send | receive | send-receive ]? [detail?] )
no trace flag ( [ route | normal | state | policy | task | timer | all | aspath ] ) | ( [ packets | open | update | keepalive | notify ] [ send | receive | send-receive ]? [detail?] )
```

**Mode**

BGP Router Configuration

**Parameters**

Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
```
state - trace state machine transition in the protocol
policy - trace the application of protocol and user-specified policy to routes being imported or exported
task - trace system interface and processing associated with this protocol
timer - trace timer usage by this protocol
all - turns on all trace flags. Note: When issuing a show command, the output will display all the trace flags rather than this all keyword.

GP-specific flags that do not allow associated actions:
- \{ aspath \} - The following BGP-specific flags cannot be associated with the send, receive, send-receive, and detail action items. These flags are defined as follows:
  - aspath - traces AS Path messages

GP-specific flags that allow associated actions:
- \{ packets | open | update | keepalive | notify \} - The following BGP-specific flags can be associated with the send, receive, send-receive, and detail action items. These flags are defined as follows:
  - packets - trace all BGP packets
  - open - trace BGP Open packets
  - update - trace BGP Update packets
  - keepalive - trace BGP Keepalive packets
  - notify - trace BGP Notification messages
  - \{ send | receive | send-receive \}? - optionally specify whether to limit the tracing to packets sent, received, or both
  - \{detail?\} - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

Description
Use the trace flag command to specify tracing flags for BGP tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both open and keepalive packets in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, trace flags specify that both the sent and received update and keepalive messages are traced in detail. This tracing information will be written to the file /var/tmp/bgp.log.

```
(config)# router bgp 65525
(config-router-bgp)# bgp router-id 1.2.3.4
(config-router-bgp)# trace file /var/tmp/bgp.log
(config-router-bgp)# trace flag update send-receive detail
(config-router-bgp)# trace flag keepalive send-receive detail
(config-router-bgp)# exit
(config)#
```
Chapter 9

Internet Control Message Protocol (ICMP)

In This Chapter

Internet Control Message Protocol (ICMP) Overview 162
router icmp 162
trace file 163
trace flag 164

Internet Control Message Protocol (ICMP) Overview

On systems without the BSD routing socket, Advanced Routing Suite listens to ICMP messages received by the system. Advanced Routing Suite currently supports redirect. Processing of ICMP redirect messages is handled by the "router redirect" command. See "Redirect Processing" for more information.

Currently, the only reason to enter ICMP Router Configuration mode is to be able to trace the ICMP messages that Advanced Routing Suite receives. These messages can be traced to a separate log file as is allowed by any Advanced Routing Suite trace file command. This allows for easy separation of non-redirect ICMP messages from redirect messages in the trace file.

router icmp

Name

router icmp - allows the user to enter ICMP Router Configuration Mode

Syntax

router icmp

Mode

Global Configuration

Parameters

none

Description

The router icmp command allows you to enter ICMP Router Configuration mode, where, currently only tracing options are supported.

Note: Because ICMP is always on, a negative form of this command is not available. Type exit at the (config-icmp)# prompt to leave ICMP Router Configuration mode.

Default

none
Internet Control Message Protocol (ICMP)

Command History
NGC 2.2 - This command was introduced.

Examples
The following example shows how to enter and exit ICMP Router Configuration mode on this router.

```
(config)# router icmp
(config-icmp)# exit
(config)#
```

trace file

Name
trace file - specifies file options during tracing for the ICMP protocol

Syntax
```
trace file file_name [|| no-timestamp || overwrite]?
no trace file file_name [ no-timestamp || overwrite]?
```

Mode
ICMP Router Configuration Mode

Parameters
- **file_name** - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
- **no-timestamp** - specifies that a timestamp should not be prepended to all trace lines
- **overwrite** - specifies to begin tracing by appending or truncating an existing file

Description
The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For ICMP, the `trace file` command in ICMP Router Configuration Mode specifies a file for tracing of all ICMP events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `trace flag` command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands. Consider the following example:

```
(config-icmp)# trace file /var/log/icmp.log no-timestamp
(config-icmp)# trace file /var/log/icmp.log
```

The option given in the second command completely replaces that given in the first.

Default
ICMP tracing is turned off by default.

Command History
NGC 2.2 - This command was introduced.
Examples

In the following example, ICMP tracing is written to the file "/var/tmp/icmp.log". No timestamp will display at the beginning of the trace lines.

```plaintext
(config)# router icmp
(config-icmp)# trace file /var/tmp/icmp.log no-timestamp
```

trace flag

Name

`trace flag` - specifies additional tracing flags for ICMP tracing

Syntax

```plaintext
trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packet | redirect | router-discovery | info | error ] [ send | receive | send-receive ]? [detail?] )

no trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packet | redirect | router-discovery | info | error ] [ send | receive | send-receive ]? [detail?] )
```

Mode

ICMP Router Configuration

Parameters

Flags common to all protocols:

```plaintext
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

ICMP-specific flags:

```plaintext
[ packet | redirect | router-discovery | info | error ] - These ICMP-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```

- **packets** - trace all ICMP packet types
- **redirect** - trace only ICMP redirect packets
- **router-discovery** - trace only ICMP router discovery packets
- **info** - trace only ICMP informational packets, which include mask request/response, info request/response, echo request/response, and time stamp request/response
- **error** - trace only ICMP error packets, which include time exceeded, parameter problem, unreachable, and source quench
- `[send | receive | send-receive]`? - optionally specify whether to limit the tracing to packets sent, received, or both
- `[detail?]` - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

**Description**

Use the `trace` flag command to specify tracing flags for ICMP tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both redirect and info packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, trace flags specify that both the sent and received request and response messages are traced in detail. This tracing information will be written to the file `/var/tmp/icmp.log`.

```
(config)# router icmp
(config-icmp)# trace file /var/tmp/icmp.log
(config-icmp)# trace flag request send-receive detail
(config-icmp)# trace flag response send-receive detail
```

**See Also**

Chapter 9 "Trace Options"
Chapter 10
Fast Open Shortest Path First (OSPF)

In This Chapter

Fast Open Shortest Path First (OSPF) Overview 168
router ospf 172
advertise-subnet 173
authentication 174
compatible rfc1583 176
dead-interval 176
distance 177
enable 178
enable-te 179
hello-interval 180
igp-shortcut 181
inherit-metric 181
monitor-auth-key 182
multicast-rib 183
network area 184
nssa-inherit-metric 185
nssa-stability-interval 186
poll-interval 186
priority 187
redistribute 188
redistribute-nssa 190
require-vbit 191
restart-allow-changes 192
restart-enable 193
restart-max-sync-time 194
restart-type 194
retransmit-interval 195
router-id 196
timers spf 197
trace file 198
trace flag 199
transmit-delay 200
area advertise-subnet 201
area authentication 202
area dead-interval 204
area filter 205
area hello-interval 206
area nssa 207
area nssa-range 208
area nssa-translate-always 209
area poll-interval 210
area priority 211
area range 212
area retransmit-interval 213
area stub 214
area stubhost 214
area stubnetwork 215
area transmit-delay 216
area virtual-link 217
default-metric 219
default-nssa-metric 219
default-nssa-type 220
default-preference 221
default-tag 222
default-type 222
advertise-subnet 223
allow-all 224
authentication 225
cost 227
dead-interval 228
enable 229
hello-interval 230
neighbor 231
network 232
no-multicast 232
passive-interface 233
poll-interval 234
priority 235
retransmit-interval 236
traffic-eng administrative-weight 237
traffic-eng attribute-flags 238
traffic-eng bandwidth 239
transmit-delay 240
ip ospf advertise-subnet 241
ip ospf allow-all 242
ip ospf area 243
ip ospf authentication 244
ip ospf cost 246
ip ospf dead-interval 246
ip ospf enable 247
ip ospf hello-interval 248
ip ospf neighbor 249
ip ospf network 250
ip ospf no-multicast 251
ip ospf passive-interface 251
ip ospf poll-interval 252
ip ospf priority 253
ip ospf retransmit-interval 254
ip ospf traffic-eng administrative-weight 255
ip ospf traffic-eng attribute-flags 256
ip ospf traffic-eng bandwidth 257
ip ospf transmit-delay 258
Fast Open Shortest Path First (OSPF) Overview

Open Shortest Path First Routing (OSPF) is a shortest path first or link-state protocol. OSPF is an interior gateway protocol that distributes routing information between routers in a single autonomous system (AS). OSPF chooses the least-cost path as the best path. OSPF is suitable for complex networks with a large number of routers because it provides equal-cost multi-path routing, where packets to a single destination can be sent via more than one interface simultaneously.

In a link-state protocol, each router maintains a database describing the entire AS topology, which it builds out of the collected link state advertisements of all routers. Each participating router distributes its local state (i.e., the router’s usable interfaces and reachable neighbors) throughout the AS by flooding. Each multi-access network that has at least two attached routers has a designated router and a backup designated router. The designated router floods a link state advertisement for the multi-access network and has other special responsibilities. The designated router concept reduces the number of adjacencies required on a multi-access network.

OSPF allows networks to be grouped into areas. Routing information passed between areas is abstracted, potentially allowing a significant reduction in routing traffic. OSPF uses the following four different types of routes, listed in order of preference: intra-area, inter-area, type 1 Autonomous System External (ASE), and type 2 ASE. Intra-area paths have destinations within the same area. Inter-area paths have destinations in other OSPF areas. Both types of ASE routes are routes to destinations external to OSPF (and usually external to the AS). Routes exported into OSPF ASE as type 1 ASE routes are supposed to be from interior gateway protocols (such as RIP) whose external metrics are directly comparable to OSPF metrics. When a routing decision is being made, OSPF will add the internal cost to the AS border router to the external metric. Type 2 ASEs are used for exterior gateway protocols whose metrics are not comparable to OSPF metrics. In this case, only the internal OSPF cost to the AS border router is used in the routing decision.

**Note:** It is possible to export a route into OSPFASE that has a nexthop inside an NSSA and not export that route into OSPFNSSA. A type 7 route (OSPFNSSA) should be originated in addition to the type 5 route (OSPFASE) in this case. To configure this, first do not originate a type 5 with a nexthop inside an NSSA. Second, originate all such routes into both OSPFASE and OSPFNSSA.

From the topology database, each router constructs a tree of the shortest paths, with itself as the root. This shortest-path tree gives the route to each destination in the AS. Externally-derived routing information appears on the tree as leaves. The link-state advertisement format distinguishes between information acquired from external sources and information acquired from internal routers, so there is no ambiguity about the source or reliability of routes. Externally-derived routing information (for example, routes learned from BGP) is passed transparently through the AS and is kept separate from OSPF’s internally derived data. Each external route can also be tagged by the advertising router, enabling routers on the borders of the AS to pass additional information between them.

OSPF optionally includes Type of Service (TOS) routing and allows administrators to install multiple routes to a given destination for each type of services, such as low delay or high throughput. A router running OSPF uses the destination address and the type of service to choose the best route to the destination.

OSPF intra- and inter-area routes are always imported into the Advanced Routing Suite routing database with a preference (or distance) of 10. Because it would violate the protocol if an OSPF router did not participate fully in the area’s OSPF, it is not possible to override this preference. Although it is possible to give other routes better preference values explicitly, doing so would violate the OSPF protocol and could lead to incorrect routing. Therefore, the only types of policy supported in OSPF are ASE import and ASE and NSSA export.
Hardware multicast capabilities are also used where possible to deliver link-status messages. OSPF areas are connected by the backbone area, the area with identifier 0.0.0.0. All areas must be logically contiguous, and the backbone is no exception. To permit maximum flexibility, OSPF allows the configuration of virtual links, which enables the backbone area to appear contiguous despite the physical reality.

Because a separate copy of the link-state algorithm is run for each area, most configuration parameters are defined on a per-area basis. All routers in an area must agree on that area’s parameters. Misconfiguration will keep neighbors from forming adjacencies between themselves, and routing information might not flow or could loop.

Advanced Routing Suite can run over a variety of physical connections: serial connections, LAN interfaces, ATM, or FDDI. The OSPF configuration supports three different types of connections in the interface clauses:

**LAN and Point-to-Point**

An example of a LAN interface is an Ethernet or a FDDI interface. A point-to-point interface can be a serial line using Point-to-Point protocol. Advanced Routing Suite will use a Multicast IP address on LAN interfaces to reach OSPF routers.

**Non-Broadcast Multiple Access**

ATM with virtual circuits is an example of a Non-Broadcast Multiple Access medium. Because there is no general multicast in all ATM devices, each router must be listed so that Advanced Routing Suite can poll each router. Advanced Routing Suite will unicast the packets to the routers in the NBMA network.

**Point-to-Multipoint**

Point-to-Multipoint connectivity is used when the network does not provide full connectivity to all routers in the network. Just as on the NBMA format, you must provide a list of routers that the Advanced Routing Suite daemon will query as OSPF peers.

Notes:

- It is suggested that users of OSPF configure interface parameters using a physical interface name (for example, "fxp0"). This provides the clearest, least ambiguous way of configuring these parameters. Using a logical address (for example, "192.168.10.1") can lead to undesirable results.
- If the same interface is configured in two different areas, the interface will run on the numerically lowest area address.
- Floating interfaces are those that are configured independently of an area and, thus, can "float" between areas. Conflicting configurations can occur between Area interface commands and Floating interface commands. For example, a user can configure the Hello interval to be 1 for an interface through the Floating interface command or to be 2 for that same interface through the Area interface command. In such a case, the Area interface command configuration is selected.

**OSPF Commands**

**Global Configuration Mode OSPF Commands**

The following command is configured in Global Configuration Mode:

"router ospf" ("router ospf" on page 172)

**Global Commands**

The following are OSPF global-wide commands that are configured in OSPF Router Configuration Mode:

"advertise-subnet" ("advertise-subnet" on page 173)
"authentication" ("authentication" on page 174)
"compatible rfc1583" ("compatible rfc1583" on page 176)
"dead-interval" ("dead-interval" on page 176)
"distance" ("distance" on page 177)
"enable" ("enable" on page 178)
"enable-te" ("enable-te" on page 179)
"hello-interval" ("hello-interval" on page 180)
"igp-shortcut" ("igp-shortcut" on page 181)
"inherit-metric" ("inherit-metric" on page 181)
"monitor-auth-key" ("monitor-auth-key" on page 182)
"multicast-rib" ("multicast-rib" on page 183)
"network area" ("network area" on page 184)
"nssa-inherit-metric" ("nssa-inherit-metric" on page 185)
"nssa-stability-interval" ("nssa-stability-interval" on page 186)
"poll-interval" ("poll-interval" on page 186)
"priority" ("priority" on page 187)
"redistribute" ("redistribute" on page 188)
"redistribute-nssa" ("redistribute-nssa" on page 190)
"require-vbit" ("require-vbit" on page 191)
"restart-allow-changes" ("restart-allow-changes" on page 192)
"restart-enable" ("restart-enable" on page 193)
"restart-max-sync-time" ("restart-max-sync-time" on page 194)
"restart-type" ("restart-type" on page 194)
"retransmit-interval" ("retransmit-interval" on page 195)
"router-id" ("router-id" on page 196)
"timers spf" ("timers spf" on page 197)
"trace file" ("trace file" on page 198)
"trace flag" ("trace flag" on page 199)
"transmit-delay" ("transmit-delay" on page 200)

Area Commands

The following are OSPF area-wide commands that are configured in OSPF Router Configuration Mode. These configurations override any similar global-wide commands:

"area advertise-subnet" ("area advertise-subnet" on page 201)
"area authentication" ("area authentication" on page 202)
"area dead-interval" ("area dead-interval" on page 204)
"area filter" ("area filter" on page 205)
"area hello-interval" ("area hello-interval" on page 206)
"area nssa" ("area nssa" on page 207)
"area nssa-range" ("area nssa-range" on page 208)
"area nssa-translate-always" ("area nssa-translate-always" on page 209)
"area poll-interval" ("area poll-interval" on page 210)
"area priority" ("area priority" on page 211)
"area range" ("area range" on page 212)
"area retransmit-interval" ("area retransmit-interval" on page 213)
"area stub" ("area stub" on page 214)
"area stubhost" ("area stubhost" on page 214)
"area stubnetwork" ("area stubnetwork" on page 215)
"area transmit-delay" ("area transmit-delay" on page 216)
"area virtual-link" ("area virtual-link" on page 217)

**Default Commands**
The following OSPF default commands are configured in OSPF Router Configuration Mode:
"default-metric" ("default-metric" on page 219)
"default-nssa-metric" ("default-nssa-metric" on page 219)
"default-nssa-type" ("default-nssa-type" on page 220)
"default-preference" ("default-preference" on page 221)
"default-tag" ("default-tag" on page 222)
"default-type" ("default-type" on page 222)

**Area Interface Commands**
The following commands are configured in Area Interface Configuration Mode:
"advertise-subnet" ("advertise-subnet" on page 223)
"allow-all" ("allow-all" on page 224)
"authentication" ("authentication" on page 225)
"cost" ("cost" on page 227)
"dead-interval" ("dead-interval" on page 228)
"enable" ("enable" on page 229)
"hello-interval" ("hello-interval" on page 230)
"neighbor" ("neighbor" on page 231)
"network" ("network" on page 232)
"no-multicast" ("no-multicast" on page 232)
"passive-interface" ("passive-interface" on page 233)
"poll-interval" ("poll-interval" on page 234)
"priority" ("priority" on page 235)
"retransmit-interval" ("retransmit-interval" on page 236)
"traffic-eng administrative-weight" ("traffic-eng administrative-weight" on page 237)
"traffic-eng attribute-flags" ("traffic-eng attribute-flags" on page 238)
"traffic-eng bandwidth" ("traffic-eng bandwidth" on page 239)
"transmit-delay" ("transmit-delay" on page 240)

**Floating Interface Commands**
The following commands are configured in Interface Configuration Mode:
"ip ospf advertise-subnet" ("ip ospf advertise-subnet" on page 241)
"ip ospf allow-all" ("ip ospf allow-all" on page 242)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf authentication" ("ip ospf authentication" on page 244)
"ip ospf cost" ("ip ospf cost" on page 246)
"ip ospf dead-interval" ("ip ospf dead-interval" on page 246)
"ip ospf enable" ("ip ospf enable" on page 247)
"ip ospf hello-interval" ("ip ospf hello-interval" on page 248)
"ip ospf neighbor" ("ip ospf neighbor" on page 249)
"ip ospf network" ("ip ospf network" on page 250)
"ip ospf no-multicast" ("ip ospf no-multicast" on page 251)
"ip ospf passive-interface" ("ip ospf passive-interface" on page 251)
"ip ospf poll-interval" ("ip ospf poll-interval" on page 252)
"ip ospf priority" ("ip ospf priority" on page 253)
"ip ospf retransmit-interval" ("ip ospf retransmit-interval" on page 254)
"ip ospf traffic-eng administrative-weight" ("ip ospf traffic-eng administrative-weight" on page 255)
"ip ospf traffic-eng attribute-flags" ("ip ospf traffic-eng attribute-flags" on page 256)
"ip ospf traffic-eng bandwidth" ("ip ospf traffic-eng bandwidth" on page 257)
"ip ospf transmit-delay" ("ip ospf transmit-delay" on page 258)

Querying Commands
"show ip ospf" ("show ip ospf" on page 259)
"show ip ospf border-routers" ("show ip ospf border-routers" on page 260)
"show ip ospf database" ("show ip ospf database" on page 260)
"show ip ospf interface" ("show ip ospf interface" on page 261)
"show ip ospf neighbor" ("show ip ospf neighbor" on page 262)
"show ip ospf request-list" ("show ip ospf request-list" on page 263)
"show ip ospf retransmission-list" ("show ip ospf retransmission-list" on page 263)
"show ip ospf summary-address" ("show ip ospf summary-address" on page 264)
"show ip ospf virtual-links" ("show ip ospf virtual-links" on page 265)

**router ospf**

**Name**

router ospf – enters configuration mode for an instance in OSPF

**Syntax**

```
router ospf instance_id
no router ospf instance_id
```

**Mode**

Global Configuration

**Parameters**

`instance_id` – an integer from 1 to 4,294,967,294, inclusive, identifying the OSPF instance that you want to configure

**Description**

Use the `router ospf` command to enter OSPF Router Configuration mode for an OSPF instance. Once you are in OSPF Router Configuration mode, you can begin configuring a new OSPF instance or change an existing configuration.

**Default**

None. You must specify this command with an `instance_id` in order to configure OSPF.
Command History
NGC 2.2 - This command was introduced.

Examples
The following example shows how to enter OSPF Router Configuration mode for OSPF instance 1.

```
(config)# router ospf 1
(config-router-ospf)#
```

advertise-subnet

Name
advertise-subnet - specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask instead of a host route to the remote IP

Syntax
```
advertise-subnet
no advertise-subnet
```

Mode
OSPF Router Configuration

Parameters
none

Description
The `advertise-subnet` command specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP. Because the netmask is sometimes set improperly on point-to-point interfaces, this option disabled by default.

This command can be overridden at the area and interface levels.

Default
If `advertise-subnet` is not specified, it is the same as if the user had specified the following:
```
(config-router-ospf)# no advertise-subnet
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns the global `advertise-subnet` on, then turns it off for area 1.2.3.4.

```
(config)# router ospf 1
(config-router-ospf)# advertise-subnet
(config-router-ospf)# no area 1.2.3.4 advertise-subnet
(config-router-ospf)# exit
(config)#
```

See Also
"area advertise-subnet" ("area advertise-subnet" on page 201)
"ip ospf advertise-subnet" ("ip ospf advertise-subnet" on page 241)

**authentication**

**Name**

`authentication` - specifies the type of OSPF authentication used and any key values

**Syntax**

```
authentication [ [simple key] | [ md5 id_number md5_key
[(start-generate date_time) ||
(stop-generate date_time) || (start-accept date_time)
(stop-accept date_time)] ]
```

```
no authentication [simple key | md5 id_number]
```

**Mode**

OSPF Router Configuration

**Parameters**

- **simple key** - specifies simple (clear password) authentication. The value for `key` is specified as a one- to eight-character string.

- **md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time) || (start-accept date_time) || (stop-accept date_time)]** - specifies the authentication used for specifying md5 cryptographic authentication. The value for `id_number` is an integer with a value between 1 and 255, inclusive. The value for `md5_key` is a one- to sixteen-character string. The `start` and `stop` values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.

**Description**

Authentication can help to guarantee that routing information is imported only from trusted routers. A variety of authentication schemes can be used, but a single scheme must be configured for each network. The use of different schemes enables some interfaces to use much stricter authentication than others. The two authentication schemes available are simple, and MD5.

The `authentication` command specifies the type of global authentication and key values used in OSPF. The negative form of this command removes authentication for the area. Authentication is used by OSPF to generate and verify the authentication field in the OSPF header. The global authentication is the default and can be overridden at the area command and interface levels.

When you want to keep certain routers from exchanging OSPF packets, use the simple form of authentication. The interfaces that the packets are to be sent on still need to be trusted, because the key will be placed in the packets and can be seen by anyone with access to the network.

When you do not trust other users of your network, use MD5 authentication. The system works by using shared secret keys. Because the keys are used to sign the packets with an MD5 checksum, they cannot be forged or tampered with. Because the keys are not included in the packet, snooping the key is not possible. Users of the network can still snoop the contents of packets, however, because the packets are not encrypted.

Advanced Routing Suite’s MD5 authentication is compliant with the specification in OSPF RFC 2328. This specification uses the MD5 algorithm and an authentication key of up to 16 characters. RFC 2328 allows multiple MD5 keys per interface. Each key has associated time ranges.

**Note:** In order to turn off authentication, you must include the authentication type in the `no` form of the command. For example, if MD5 authentication was configured, then simply specifying `no authentication` will not work. In addition, you must specify the `simple key` when turning off simple authentication, and you must specify the MD5 `id_number` when turning off MD5 authentication.
Default
The default is for no authentication to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures simple authentication in OSPF instance 1.

```
(config)# router ospf 1
(config-router-ospf)# authentication simple abc
(config-router-ospf)# exit
(config)#
```

Example 2
The following example configures md5 authentication for OSPF instance 2. The ID for this md5 configuration is 1, and the key is configured as "bar".

```
(config)# router ospf 2
(config-router-ospf)# authentication md5 1 bar
(config-router-ospf)# exit
(config)#
```

Example 3
The following example turns off the MD5 authentication that was configured in Example 2. Note that specifying the MD5 key is not required, but specifying the MD5 ID is.

```
(config)# router ospf 2
(config-router-ospf)# no authentication md5 1
(config-router-ospf)# exit
(config)#
```

Example 4
The following example configures md5 authentication for OSPF instance 3. The md5 authentication is configured with a start-generate time set to January 02, 2004 at 21:30 hours, a stop-generate time set to January 02, 2004 at 21:45 hours, a start-accept time set to January 02, 2004 at 21:00 hours, and a stop-accept time set to January 02, 2004 at 22:00 hours.

```
(config)# router ospf 3
(config-router-ospf)# authentication md5 2 md5
    start-generate 2004-01-02.21.30
    stop-generate 2004-01-02.21.45
    start-accept 2004-01-02.21.00
    stop-accept 2004-01-02.22.00
(config-router-ospf)# exit
(config)#
```

See Also
"area authentication" ("area authentication" on page 202)
"ip ospf authentication" ("ip ospf authentication" on page 244)
### compatible rfc1583

**Name**

`compatible rfc1583` - specifies to run in RFC 1583 mode instead of RFC 2328 mode

**Syntax**

```
compatible rfc1583
no compatible rfc1583
```

**Mode**

OSPF Router Configuration

**Parameters**

none

**Description**

The `compatible rfc1583` command specifies to run in RFC 1583 mode instead of RFC 2328 mode. Do not specify this command if all the routers using an OSPF implementation in your domain are based on RFC 2328 or later. This option should be specified the same way on all routers in the domain. If any of the routers do not have this option, you should always enable this. When disabled, the preference rules for best route election are changed to eliminate certain kinds of possible routing loops.

The negative of this command, `no compatible rfc1583`, removes the requirement to run in RFC 1583 mode and reverts to RFC 2328 mode.

**Default**

If `compatible rfc1583` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# compatible rfc1583
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example disables RFC 1583 mode in for instance 1 of OSPF.

```
(config)# router ospf 1
(config-router-ospf)# no compatible rfc1583
(config-router-ospf)# exit
(config)#
```

### dead-interval

**Name**

`dead-interval` - specifies the number of seconds that can elapse without receiving a router’s hello packets before the router’s neighbors will declare it down

**Syntax**

```
dead-interval time-seconds
no dead-interval time-seconds?
```
Mode
OSPF Router Configuration

Parameters
time-seconds – an integer between 1 and 65535, inclusive, specifying a time in seconds

Description
Use the global dead-interval command to specify the number of seconds that may elapse without receiving a router's hello packets before the router's neighbors will declare it down. This value can be overridden at the area and interface levels. A general rule is for this value to equal three times the HELLO interval. Do not set this value to be less than the HELLO interval or convergence will not occur.

The negative form of this command, no dead-interval, removes the configured value and returns this to its default value of 40 seconds. Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If dead-interval is not specified, it is the same as if the user had specified the following:

(config-router-ospf)# dead-interval 40

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a global dead-interval value of 30. This value is then overridden in area 1.2.3.4 to be 25.

(config)# router ospf 1
(config-router-ospf)# dead-interval 30
(config-router-ospf)# area 1.2.3.4 dead-interval 25
(config-router-ospf)# exit
(config)#

See Also
"area dead-interval" ("area dead-interval" on page 204)
"ip ospf dead-interval" ("ip ospf dead-interval" on page 246)

distance

Name
distance – specifies how active routes that are learned from OSPF internal reachability will be selected, compared to other protocols

Syntax
distance int_value
no distance int_value?

Mode
OSPF Router Configuration
Parameters

*int_value* - the Advanced Routing Suite preference for internal routes. This can be an integer from 1 to 255, inclusive.

Description

The `distance` command specifies how active routes that are learned from the OSPF internal reachability (compared to other protocols) will be selected. When a route has been learned from more than one protocol, the active route will be selected from the protocol with the lowest distance. Each protocol has a default distance in this selection.

The negative form of this command, `no distance`, removes the configured value and returns this to its default value of 10.

Default

If `distance` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# distance 10
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the default distance for internal routes to be 100.

```
(config)# router ospf 1
(config-router-ospf)# distance 100
(config-router-ospf)# exit
(config)#
```

enable

Name

*enable* - enables an OSPF instance

Syntax

```
enable
no enable
```

Mode

OSPF Router Configuration

Parameters

none

Description

The `enable` command enables the state for an OSPF instance. The negative form of this command, `no enable`, disables the instance. If an enabled instance is disabled, then it is stopped, and all running state is deleted. All configuration state, however, is preserved. If a disabled instance is enabled, it will start running again with its preserved configuration state.

Default

If `enable` is not specified, it is the same as if the user had specified the following:
(config-router-ospf)# enable

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example disables OSPF instance 3.

```bash
(config)# router ospf 3
(config-router-ospf)# no enable
(config-router-ospf)# exit
(config)#
```

default

---

enab-te

**Name**
enab-te - causes OSPF to originate Traffic Engineering (TE) information in Opaque LSAs

**Syntax**

- `enable-te`
- `no enable-te`

**Mode**
Global Configuration

**Parameters**
none

**Description**
Use the `enable-te` command to cause OSPF to originate TE information in Opaque LSA for all running OSPF interfaces. This information does not affect normal OSPF routing and provides extra parameters for links and routers in the topology, such as available bandwidth.

**Default**
If `enable-te` is not specified, it is the same as if the user had specified:

```bash
(config)# no enable-te
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example enables Traffic Engineering LSAs for OSPF router instance 1.

```bash
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)#
```
**hello-interval**

**Name**
hello-interval - specifies the time in seconds between hello packets that the other router sends on an interface

**Syntax**
```
hello-interval time-seconds
no hello-interval time-seconds?
```

**Mode**
OSPF Router Configuration

**Parameters**
time-seconds - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

**Description**
The `hello-interval` command specifies the time in seconds between hello packets that the router sends on the interface. This option is specified here at the global level. It can be overridden in the area and interface levels. (See "area hello-interval" ("area hello-interval" on page 206) and "ip ospf hello-interval" ("ip ospf hello-interval" on page 248).)

The negative form of this command, `no hello-interval`, removes the configured value and returns this to its default value of 10 seconds. **Note: Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.**

**Default**
If `hello-interval` is not specified, it is the same as if the user had specified the following:
```
(config-router-ospf)# hello-interval 10
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures a global hello interval of 15 seconds. This is then overridden in the area statement, where it is configured to be 5 seconds.
```
(config)# router ospf 1
(config-router-ospf)# hello-interval 15
(config-router-ospf)# area 1.2.3.4 hello-interval 5
(config-router-ospf)# exit
(config)#
```

**See Also**
"area hello-interval" ("area hello-interval" on page 206)
"ip ospf hello-interval" ("ip ospf hello-interval" on page 248)
igp-shortcut

Name
igp-shortcut - enables the IGP shortcut feature for OSPF

Syntax
    igp-shortcut
    no igp-shortcut

Mode
OSPF Router Configuration

Parameters
none

Description
The igp-shortcut command enables the IGP Shortcut feature for OSPF. When configured, OSPF will consider MPLS tunnels as unidirectional, directly connected, point-to-point links. To disable this feature, use the negative form of the command.

Default
If the igp-shortcut command is not explicitly configured, it is as if the user had configured the following:

    (config-router-ospf)# no igp-shortcut

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following command enables the IGP Shortcut feature for OSPF instance 1.

    (config)# router ospf 1
    (config-router-ospf)# igp-shortcut
    (config-router-ospf)# exit
    (config)#

Example 2
The following command disables the IGP Shortcut feature for OSPF instance 1.

    (config)# router ospf 1
    (config-router-ospf)# no igp-shortcut
    (config-router-ospf)# exit
    (config)#

inherit-metric

Name
inherit-metric - configures an OSPF ASE route to inherit the metric of the external route when no metric is specified on the export policy
Syntax

\[ \text{inherit-metric} \]
\[ \text{no inherit-metric} \]

Mode

OSPF Router Configuration

Parameters

none

Description

Use the inherit-metric command to allow an OSPF ASE route to inherit the metric of the external route when no metric is specified on the export policy. This feature maintains compatibility with all the current export functions. A metric specified on the export policy takes precedence. The metric specified in the defaults section ("default-metric" ("default-metric" on page 219)) will be used if this command is not specified.

Default

If inherit-metric is not specified, it is the same as if the user had specified the following:

\[ (\text{config-router-ospf})\# \text{no inherit-metric} \]

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures ASE routes exported into OSPF to inherit the metric of the external route.

\[ (\text{config})\# \text{router ospf 1} \]
\[ (\text{config-router-ospf})\# \text{inherit-metric} \]
\[ (\text{config-router-ospf})\# \text{exit} \]
\[ (\text{config})\# \]

monitor-auth-key

Name

monitor-auth-key - sets an authentication key when using the OSPF Monitor tool

Syntax

\[ \text{monitor-auth-key} \text{ string} \]
\[ \text{no monitor-auth-key} \text{ string} \]

Mode

OSPF Router Configuration

Parameters

string - a string of up to eight characters
**Description**

Use the `monitor-auth-key` command to set an authentication key when using the OSPF Monitor tool. The negative form of this command, `no monitor-auth-key`, removes the password previously defined by this command and reverts to the default of no password.

**Default**

If `monitor-auth-key` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no monitor-auth-key
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example sets the `monitor-auth-key` authentication to "abc-auth".

```
(config)# router ospf 2
(config-router-ospf)# monitor-auth-key abc-auth
(config-router-ospf)# exit
(config)#
```

**See Also**

"Chapter 4 OSPF Monitor (ospfmon)" in Operating Advanced Routing Suite

---

**multicast-rib**

**Name**

`multicast-rib` - specifies the RIB in which OSPF internal routes are installed

**Syntax**

```
multicast-rib
no multicast-rib
```

**Mode**

OSPF Router Configuration

**Parameters**

none

**Description**

The `multicast-rib` command specifies the RIB into which OSPF internal routes are installed. The unicast RIB is required and is the default. In code bases that support extended RIBs, OSPF routes can be installed in the multicast RIB.

This command has no effect on self-originated ASE or NSSA routes because they are exported from another protocol. The negative form of this command, `no multicast-rib` reverts this back to its default of `no multicast-rib`.

**Default**

If `multicast-rib` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no multicast-rib
```
Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures OSPF routes to be installed in the unicast and multicast RIBs.

```
(config)# router ospf 1
(config-router-ospf)# multicast-rib
(config-router-ospf)# exit
(config)#
```

**network area**

**Name**

`network area` - defines the interfaces on which OSPF will run, and defines the area ID for those interfaces

**Syntax**

```
network ip_address wildcard-mask area area_id
no network ip_address wildcard-mask area area_id
```

**Mode**

OSPF Router Configuration

**Parameters**

- `ip_address` - a valid IP address
- `wildcard-mask` - IP-address-type mask that includes "don't care" bits
- `area_id` - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of `0.0.0.0` signals that the area is a backbone. If you intend to associate areas with IP subnets, you can specify a subnet address for this value.

**Description**

Use the `network area` command to specify interfaces on which OSPF runs and to define the area ID for those interfaces. Use the negative form of this command, `no network area`, to disable OSPF routing for any specified interfaces.

**Default**

This command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples

**Example 1**

The following example defines network 192.168.10.5 0.0.0.255 in area 1.2.3.4

```
(config)# router ospf 1
(config-router-ospf)# network 192.168.10.5 0.0.0.255 area 1.2.3.4
(config-router-ospf)# exit
```
Example 2
The following example disables OSPF routing on interface 102.168.10.10 in area 1.2.3.4.

```
(config)# router ospf 1
(config-router-ospf)# no network 192.168.10.10 0.0.0.0 area 1.2.3.4
(config-router-ospf)# exit
(config)#
```

**nssa-inherit-metric**

**Name**

nssa-inherit-metric - allows an OSPF NSSA route to inherit the metric of the external route when no metric is specified on the export policy

**Syntax**

```
nssa-inherit-metric
no nssa-inherit-metric
```

**Mode**

OSPF Router Configuration

**Parameters**

none

**Description**

The nssa-inherit-metric command allows an OSPF NSSA route to inherit the metric of the external route when no metric is specified on the export policy. This feature maintains compatibility with all the current export functions. A metric specified on the export policy will take precedence.

**Default**

If nssa-inherit-metric is not specified, it is the same as if the user had specified the following:

```
(config)# no nssa-inherit-metric
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures NSSA routes exported into OSPF to inherit the metric of the external route.

```
(config)# router ospf 1
(config-router-ospf)# nssa-inherit-metric
(config-router-ospf)# exit
(config)#
```
**nssa-stability-interval**

**Name**

*nssa-stability-interval* - sets the length of time in seconds that an NSSA translator will continue to translate after losing the translator election

**Syntax**

```plaintext
nssa-stability-interval time-seconds
no nssa-stability-interval time-seconds?
```

**Mode**

OSPF Router Configuration

**Parameters**

*time-seconds* - an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

**Description**

An NSSA ABR can translate Type 7 LSAs into Type 5 LSAs. A translator election is run to determine which one ABR for a given NSSA will translate. When an ABR was a translator and loses this election, it will cease translating after the amount of time specified by *time-seconds*. This allows the newly translated Type 5 LSAs from the new translator to be flooded throughout the domain before the currently translated Type 5 LSAs from this translator are flushed and the Type 5 LSAs resulting from direct translation are allowed to age out.

**Default**

If *nssa-stability-interval* is not specified, it is the same as if the user had specified the following:

```plaintext
(config-router-ospf)# nssa-stability-interval 40
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example sets the NSSA stability interval to 10 seconds.

```plaintext
(config)# router ospf 1
(config-router-ospf)# nssa-stability-interval 10
(config-router-ospf)# exit
(config)#
```

**poll-interval**

**Name**

*poll-interval* - specifies the length of time, in seconds, between OSPF packets that the router send before adjacency is established with a neighbor

**Syntax**

```plaintext
poll-interval time-seconds
no poll-interval time-seconds?
```

**Mode**

OSPF Router Configuration
Parameters

- time-seconds - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

Description

The `poll-interval` command specifies the length of time, in seconds, between OSPF packets that the router sends before an adjacency is established with a neighbor. Utilizing this command reduces network overhead in cases where a router may have a neighbor on a given interface at the expense of initial convergence time.

This value is configured here at the global level and can be overridden at the area and interface levels. The negative form of this command, `no poll-interval`, removes the configured time-seconds value and returns this to its default value of 120 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `poll-interval` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# poll-interval 120
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a global poll interval of 100 seconds. This value is overridden in interface fxp1, where it is configured to be 110 seconds.

```
(config)# router ospf 1
(config-router-ospf)# poll-interval 100
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf poll-interval 110
(config-if)# exit
(config)#
```

See Also

"area poll-interval" ("area poll-interval" on page 210)
"ip ospf poll-interval" ("ip ospf poll-interval" on page 252)

priority

Name

- priority - specifies the priority for becoming the designated router (DR)

Syntax

```
priority level
no priority level?
```

Mode

OSPF Router Configuration
**Parameters**

*level* - a priority number between 0 and 255, inclusive, for becoming a DR

**Description**

The `priority` command specifies a number between 0 and 255 that configures the priority for becoming the DR. When more than one router attached to a network attempts to become the DR, the one with the highest priority wins. If the competing routes have the same priority, the one with the highest router ID becomes the DR. The router coming in second in the election becomes the backup DR. A router with a router priority set to 0 is ineligible to become the DR.

This value is specified here at the global level and can be overridden at the area and interface levels ("area priority" ("area priority" on page 211) and "ip ospf priority" ("ip ospf priority" on page 253)). The negative form of this command, `no priority`, removes the configured value and returns this to its default value of 0. **Note:** Specifying a value for `level` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `priority` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# priority 1
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a global priority of 10. This value is then overridden in area 1.2.3.4, where it is configured to be 15.

```
(config)# router ospf 1
(config-router-ospf)# priority 10
(config-router-ospf)# area 1.2.3.4 priority 15
(config-router-ospf)# exit
(config)#
```

**See Also**

"area priority" ("area priority" on page 211)  
"ip ospf priority" ("ip ospf priority" on page 253)

---

**redistribute**

**Name**

redistribute - specifies routes to export to OSPF

**Syntax**

```
redistribute protocol [route-map name]? {0,9}  
no redistribute protocol [route-map name]?
```

**Mode**

OSPF Router Configuration
Parameters

protocol – the protocol name whose routes you want to redistribute to the current protocol being configured. Valid protocols are aggregate, bgp, direct, isis, kernel, ospf, ospf-ase, rip, and static.

route-map name – the name of a route map to apply to these routes. Specifying this is optional.

{0,9} – although this command can be given multiple times, it can only be given once for each of the nine configurable protocols. In other words, if a redistribute command is given for a protocol and route map, and then given again for the same protocol with a different route map, the second configuration overrides the first.

Description

Use the redistribute command to specify routes to export into OSPF. This command causes routes from the specified protocol to be considered for redistribution into the current protocol. Additionally, if a route map is specified, then routes from the specified protocol that match the named route map will be considered for redistribution into the current protocol. If the referenced route map has not yet been configured, then an empty route map is created with the specified name.

Note: Configuring this away from its default removes the implicitly configured default. You will have to go back and specify to redistribute OSPF and direct routes after the first redistribute configuration in order to export those routes.

Default

The default is to redistribute OSPF and the direct routes associated with the interfaces on which OSPF is running. Note that this is an implicit default that is wiped away with the first redistribute configuration.

Example

Example 1

In the following example OSPF instance 2 is configured to redistribute all BGP and RIP routes.

```
(config)# router ospf 2
(config-router-ospf)# redistribute bgp
(config-router-ospf)# redistribute rip
(config-router-ospf)# exit
(config)#
```

Example 2

The following example configures a community set, "set1", that permits AS: num 101:102. It then configures an extended community set "ext-set1", that permits Route Target AS: num 201:202.

```
(config)# ip community-set set1 permit 101:102
(config)# ip extcommunity-set ext-set1 permit rt 201:202
```

The two are then added to a community list, called "commlist1".

```
(config)# ip community-list commlist1 permit set1
(config)# ip community-list commlist1 permit ext-set1
```

The community list is then applied to a route map called "match-commlist1". If the route map matches BGP Community list "commlist1", then the metric for routes will be set to 20.

```
(config)# route-map match-commlist1
(config-route-map)# match community commlist1
(config-route-map)# set metric 20
(config-route-map)# exit
(config)#
```

Finally, the route map ("match-commlist1") is applied to BGP routes and exported into instance 1 of OSPF.
(config)# router ospf 1
(config-router-ospf)# redistribute bgp route-map match-commlist1
(config-router-ospf)# exit
(config)#

**Example 3**

In the following example, route map "abc" is configured with the following match criteria:

If a route matches interface "fxp1" and a pre-configured BGP Community labeled "bgpcomm1", then communities specified in community "com-set-1" will be added to the route, communities specified in community labeled "com-set-2" will be deleted from the route, and the metric of the route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match community-set bgpcomm1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
```

This route map is then applied to static routes and exported into OSPF instance 2.

```
(config)# router ospf 2
(config-router-ospf)# redistribute static route-map abc
(config-router-ospf)# exit
```

### redistribute-nssa

**Name**

`redistribute-nssa` - specifies routes to export to OSPF NSSA in an OSPF instance

**Syntax**

```
redistribute-nssa protocol [route-map name]? {0,9}
no redistribute-nssa protocol [route-map name]?
```

**Mode**

OSPF Router Configuration

**Parameters**

- `protocol` - the protocol name whose routes you want to redistribute to the current protocol being configured. Valid protocols are aggregate, bgp, direct, kernel, isis, ospf, ospf-ase, rip, and static.

- `route-map name` - the name of a route map to apply to these routes. Specifying this is optional.

- `{0,9}` - although this command can be given multiple times, it can only be given once for each of the nine configurable protocols. In other words, if a `redistribute` command is given for a protocol and route map, and then given again for the same protocol with a different route map, the second configuration overrides the first.

**Description**

Use the `redistribute-nssa` command to specify routes to export into all NSSA areas in the current OSPF instance. This command causes routes from the specified `protocol` to be considered for redistribution.
into OSPF-NSSA. Additionally, if a route map is specified, then routes from the specified protocol that match the named route map will be considered for redistribution into OSPF-NSSA.

If the referenced route map has not yet been configured, then an empty route map is created with the specified name.

**Default**

The default is that OSPF does not redistribute any routes into NSSA.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, a route map "abc" is configured to match all IPv4 prefixes (configured in prefix list "pl1"), to set the exported metric to 1, and to set the NSSA Propagate (P) bit.

```
(config)# ip prefix-list pl1 seq 1 permit 0.0.0.0/0 le 32
(config)# route-map abc
(config-route-map)# match ip address prefix-list pl1
(config-route-map)# set metric 1
(config-route-map)# set propagate
(config-route-map)# exit
(config)#
```

This route map is then applied to static routes and exported into OSPF NSSA.

```
(config)# router ospf 1
(config-router-ospf)# redistribute-nssa static route-map abc
(config-router-ospf)# exit
(config)#
```

**require-vbit**

**Name**

require-vbit – instructs the SPF code to require that the vbit be set in the router LSAs of routers that are the end points of virtual links

**Syntax**

```
require-vbit
no require-vbit
```

**Mode**

OSPF Router Configuration

**Parameters**

none

**Description**

Specifying this command instructs the SPF code to require that the vbit be set in the router LSAs of routers that are the end points of virtual links. When this command is not specified, the vbit can be clear, and the virtual link will still be considered.

This command exists to work around a bug in some OSPF implementations.
Default

If `require-vbit` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no require-vbit
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example turns `require-vbit` on.

```
(config)# router ospf 1
(config-router-ospf)# require-vbit
(config-router-ospf)# exit
(config)#
```

**restart-allow-changes**

Name

`restart-allow-changes` - specifies whether changes in the network will cause any helper attempts to fail

Syntax

```
restart-allow-changes
no restart-allow-changes
```

Mode

OSPF Router Configuration

Parameters

none

Description

The `restart-allow-changes` command specifies whether a change in the network (for example, something that causes an SPF) will cause any helper attempts to fail. This command can be given for multiple instances.

Turning this command on configures the OSPF instance to cancel helper mode on all neighbors currently attempting a restart (either Graceful or Signaled) when a topology change occurs.

The negative form of this command, `no restart-allow-changes` configures the restart helper mode to ignore topology change. Note that this can cause black holes to occur in the network.

Default

If `restart-allow-changes` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no restart-allow-changes
```

Command History

NGC 2.3 - This command was introduced.

Examples

The following example turns the allow changes helper flag on.
(config)# router ospf 1
(config-router-ospf)# restart-allow-changes
(config-router-ospf)# exit

**restart-enable**

**Name**

restart-enable - specifies to attempt a restart after the next startup

**Syntax**

restart-enable

no restart-enable

**Mode**

OSPF Router Configuration

**Parameters**

none

**Description**

This command configures the restart ability for an OSPF instance. If the restart-enable command is configured for an instance when OSPF starts, and/or if Advanced Routing Suite is restarted with the -r flag (see "The Command-line Options" in Operating Advanced Routing Suite), then OSPF will perform restart operations.

Because of the nature of the CLI, this command must be given during a "conf file replace" in order for it to apply to the current startup attempt. The reason is that all OSPF configuration must be in place before any of the configuration is committed. This can only be guaranteed with a conf file replace.

**Notes:**

- A reconfiguration will not result in an attempted restart. Only occurrences that force the instance to be deleted and re-created will result in a restart.
- This command overrides the -r flag given during a restart. For example, if no restart-enable is configured for an instance, and then Advanced Routing Suite is restarted with the -r flag, restarts will not take place.

**Default**

If the restart-enable command is not specified, it is the same as if the user had specified the following:

(config-router-ospf)# no restart-enable

**Command History**

NGC 2.3 - This command was introduced.

**Examples**

The following example configures OSPF to perform restart operations.

(config)# router ospf 1
(config-router-ospf)# restart-enable
(config-router-ospf)# exit
(config)#
**restart-max-sync-time**

*Name*

`restart-max-sync-time` - configures the grace period in seconds for a restart attempt, for an OSPF instance

*Syntax*

```
restart-max-sync-time time-seconds
no restart-max-sync-time time-seconds?
```

*Mode*

OSPF Router Configuration

*Parameters*

`time-seconds` - an integer from 1 to 65535, inclusive.

*Description*

The `restart-max-sync-time` command specifies the number of seconds that elapse before a restart attempt will time out and fail. This command can be configured for multiple instances. For graceful restart, this value represents the grace period. For signaled restart, if all adjacencies are not restored by this number of seconds after the restart attempt begins, then the attempt will fail.

**Note:** Either the “restart-enable” ("restart-enable" on page 193) command must be configured as "on" or Advanced Routing Suite must be restarted with the `-r` flag (see "The Command-line Options" in Operating Advanced Routing Suite) in order for this command to take effect.

The negative form of this command, `no restart-max-sync-time`, removes the configured `time-seconds` value. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

*Default*

If `restart-max-sync-time` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# restart-max-sync-time 40
```

*Command History*

NGC 2.3 - This command was introduced.

*Examples*

The following example configures the period for a restart to be 30 seconds.

```
(config)# router ospf 1
(config-router-ospf)# restart-enable
(config-router-ospf)# restart-max-sync-time 30
(config-router-ospf)# exit
```

*See Also*

"restart-enable" ("restart-enable" on page 193)

---

**restart-type**

*Name*

`restart-type` - specifies whether the type of future restarts will be graceful or signaled
Syntax

```
restart-type [graceful | signaled]
no restart-type [graceful | signaled]?
```

Mode

OSPF Router Configuration

Parameters

```
graceful | signaled - specifies either graceful or signaled restart
```

Description

The `restart-type` command specifies the type of restart that will be attempted for future restarts. This command can be given for multiple instances. For example, one instance can perform a graceful restart and one instance can perform a signaled restart, each at the same time.

**Note:** Either the “restart-enable” ("restart-enable" on page 193) command must be configured as "on" or Advanced Routing Suite must be restarted with the `-r` flag (see "The Command-line Options" in Operating Advanced Routing Suite) in order for this command to take effect.

Default

If the `restart-type` command is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# restart-type graceful
```

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures a restart type of signaled for OSPF instance 3.

```
(config)# router ospf 3
(config-router-ospf)# restart-enable
(config-router-ospf)# restart-type signaled
(config-router-ospf)# exit
(config)#
```

See Also

"restart-enable" ("restart-enable" on page 193)

---

retransmit-interval

**Name**

`retransmit-interval` - specifies the number of seconds between link state advertisement (LSA) retransmissions for adjacencies

**Syntax**

```
retransmit-interval time-seconds
no retransmit-interval time-seconds?
```

**Mode**

OSPF Router Configuration
Parameters

time-seconds - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

Description

The retransmit-interval command sets the default for the number of seconds between LSA retransmissions for adjacencies. If a Link State Protocol (LSP) is not acknowledged within the number of seconds specified here, it is re-sent.

This command is specified here at the global level and can be overridden in the area and interface levels ("area retransmit-interval" ("area retransmit-interval" on page 213) and "ip ospf retransmit-interval" ("ip ospf retransmit-interval" on page 254)).

The negative form of this command, no retransmit-interval, removes the configured retransmit value and returns this to its default value of 5 seconds. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If retransmit-interval is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# retransmit-interval 5
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the global retransmit interval to be 10 seconds. This value is then overridden in area 1.2.3.4, where it is configured to be 15 seconds.

```
(config)# router ospf 1
(config-router-ospf)# retransmit-interval 10
(config-router-ospf)# area 1.2.3.4 retransmit-interval 15
(config-router-ospf)# exit
(config)#
```

See Also

"area retransmit-interval" ("area retransmit-interval" on page 213)
"ip ospf retransmit-interval" ("ip ospf retransmit-interval" on page 254)

**router-id**

Name

router-id - sets the OSPF 32-bit router ID for the specified instance

Syntax

```
router-id rid_value
no router-id rid_value
```

Mode

OSPF Router Configuration

Parameters

rid_value - a 32-bit address in dotted-quad notation
Description

The `router-id` command sets the OSPF 32-bit router ID for the specified instance. If the Router ID is not specified, then the instance uses the global default router ID. The negative form of this command, `no router-id`, removes the configured router ID and returns this to its default value specified by the globally defined Router ID.

Default

If `router-id` is not specified, the instance uses the globally specified Router ID. If that is not specified, then the highest IP address will be used.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the OSPF router ID to be 4.3.2.1.

```
(config)# router ospf 1
(config-router-ospf)# router-id 4.3.2.1
(config-router-ospf)# exit
(config)#
```

timers spf

Name

`timers spf` - specifies the minimum time between when OSPF receives a topology change and when it starts the SPF computation

Syntax

```
timers spf time-seconds
no timers spf time-seconds?
```

Mode

OSPF Router Configuration

Parameters

`time-seconds` - the minimum interval between SPF calculations, specified as an integer between 1 and 65535, inclusive

Description

Use the `timers spf` command to specify the minimum time between when OSPF receives a topology change and when it starts the SPF computation. The negative form of this command, `no timers spf`, removes the configured time value and returns this to its default value of 5.

Default

If `timers spf` is not specified, it is the same as if the user had specified the following:

```
(config)# timers spf 5
```

Command History

NGC 2.2 - This command was introduced.
Examples
The following example configures the SPF timer interval to be 15.

```
(config)# router ospf 1
(config-router-ospf)# timers spf 15
(config-router-ospf)# exit
(config)#
```

trace file

Name
`trace file` - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed.

Syntax
```
trace file file_name [ no-timestamp || overwrite]?
no trace file file_name [no-timestamp || overwrite]?
```

Mode
OSPF Router Configuration

Parameters
`file_name` - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
`no-timestamp` - specifies that a timestamp should not be prepended to all trace lines
`overwrite` - specifies to begin tracing by appending or truncating an existing file

Description
The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For OSPF, the `trace file` command in OSPF Router Configuration Mode specifies a file for tracing of all OSPF events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `trace flag` command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands. Consider the following example:

```
(config-router-ospf)# trace file /var/log/ospf.log
no-timestamp
(config-router-ospf)# trace file /var/log/ospf.log
```

The option given in the second command completely replaces that given in the first.

Default
OSPF tracing is turned off by default.

Command History
NGC 2.2 - This command was introduced.
Examples
In the following example, OSPF tracing is written to the file "/var/tmp/ospf.log". No timestamp will display at the beginning of the trace lines.

```
(config)# router ospf 1
(config-router-ospf)# trace file /var/tmp/ospf.log no-timestamp
```

trace flag

Name
`trace flag` - specifies OSPF-specific tracing options as well as options that are common across all protocols

Syntax
```
trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ db | ospf-state | drelect | spf | flood | debug ] ) | ( [ packets | hello | dd | lsr | lsu | lsa ] [ send | receive | send-receive ]? [ detail? ] )
```
```
o trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ db | ospf-state | drelect | spf | flood | debug ] ) | ( [ packets | hello | dd | lsr | lsu | lsa ] [ send | receive | send-receive ]? [ detail? ] )
```

Mode
OSPF Router Configuration

Parameters
Flags common to all protocols:
```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```
- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags
```

SPF-specific flags that do not allow associated actions:
```
[ ospf-state | drelect | db | spf | flood | debug ] - These OSPF-specific flags cannot be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```
- **ospf-state** - trace OSPF state change information
- **drelect** - trace the Designated Router operations
- **db** - trace the link-state database operations
- **spf** - trace the Shortest Path First (SPF) calculations
- **flood** - trace the flooding procedure
- **debug** - trace OSPF at the debugging level of detail

SPF-specific flags that allow associated actions:

```
[ packets | hello | dd | lsr | lsu | lsa ]
```

These OSPF-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:

- **packets** - trace all OSPF link-state packets
- **hello** - trace OSPF hello packets, which are used to determine neighbor reachability
- **dd** - trace OSPF Database Description (DD) packets, which are used in synchronizing OSPF databases
- **lsr** - trace OSPF link-state request packets, which are used in synchronizing OSPF databases
- **lsu** - trace OSPF link-state update packets, which are used in synchronizing OSPF databases
- **lsa** - trace OSPF link-state acknowledgement packets, which are used in synchronizing OSPF databases

```
[ send | receive | send-receive ]?
```

optionally specify whether to limit the tracing to packets sent, received, or both

```
[detail?]
```

optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

**Description**

Use the `trace flag` command to specify tracing flags for OSPF tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both task and policy packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, trace flags specify that both the sent and received link state request and hello messages are traced in detail. This tracing information will be written to the file `/var/tmp/ospf.log`.

```
(config)# router ospf 1
(config-router-ospf)# trace file /var/tmp/ospf.log
(config-router-ospf)# trace flag lsr send-receive detail
(config-router-ospf)# trace flag hello send-receive detail
```

## transmit-delay

**Name**

`transmit-delay` - specifies the time in seconds required to transmit a link state update

**Syntax**

```
transmit-delay time-seconds
no transmit-delay time-seconds?
```

**Mode**

OSPF Router Configuration
**Parameters**

*time-seconds* - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

**Description**

The `transmit-delay` command sets the estimated number of seconds required to transmit a link state update. This command takes into account transmission and propagation delays and must be greater than 0. The transmit delay is specified here at the global level. It can be overridden at the area and interface levels ("area transmit-delay" ("area transmit-delay" on page 216) and "ip ospf transmit-delay" ("ip ospf transmit-delay" on page 258)) as well as within a virtual link ("area virtual-link" ("area virtual-link" on page 217)).

The negative form of this command, `no transmit-delay`, removes the configured *time-seconds* value and returns this to its default value of 1. **Note:** Specifying a value for *time-seconds* in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `transmit-delay` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# transmit-delay 1
```

**Example**

The following example configures a global transmit delay of 5 seconds. This value is then overridden in area 1.2.3.4 to be 3 seconds.

```
(config)# router ospf 1
(config-router-ospf)# transmit-delay 5
(config-router-ospf)# area 1.2.3.4 transmit-delay 3
(config-router-ospf)# exit
```

---

**area advertise-subnet**

**Name**

`area advertise-subnet` - specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask instead of a host route to the remote IP

**Syntax**

```
area area_id advertise-subnet
no area area_id advertise-subnet
```

**Mode**

OSPF Router Configuration

**Parameters**

*area_id* - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

**Description**

The `area advertise-subnet` command specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP. Because the netmask is sometimes set improperly on point-to-point interfaces, this option is disabled by default.

Configuring this command in an area overrides a global `advertise-subnet` command. Similarly, this command can be overridden in an interface command.
Default
If area advertise-subnet is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no area area_id advertise-subnet
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns advertise-subnet on for area 1.2.3.4 and off for interface fxp1 in the same area.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 advertise-subnet
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# no ip ospf advertise-subnet
(config-if)# exit
(config)#[/code]
```

See Also
"advertise-subnet" ("advertise-subnet" on page 173)
"ip ospf advertise-subnet" ("ip ospf advertise-subnet" on page 241)

---

**area authentication**

**Name**
area authentication - specifies the type of OSPF authentication used and any key values

**Syntax**

```
area area_id authentication [ [simple key] | md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time)] || (start-accept date_time) || (stop-accept date_time)]? ]
no area area_id authentication [ simple key | md5 id_number]
```

**Mode**
OSPF Router Configuration

**Parameters**

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

simple key - specifies simple (clear password) authentication. The value for key is specified as a one- to eight-character string.

md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time)] || (start-accept date_time) || (stop-accept date_time)] - specifies the authentication used for specifying md5 cryptographic authentication. The value for id_number is an integer with a value between 1 and 255, inclusive. The value for md5_key is a one- to sixteen-character string. The start and stop values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.
Description
Authentication can help to guarantee that routing information is imported only from trusted routers. A variety of authentication schemes can be used, but a single scheme must be configured for each network. The use of different schemes enables some interfaces to use much stricter authentication than others. The two authentication schemes available are simple, and MD5.

The `area authentication` command specifies the type of authentication and key values used in OSPF. The negative form of this command removes authentication for the area. Authentication is used by OSPF to generate and verify the authentication field in the OSPF header. The global authentication is the default and can be overridden here in the area command. Similarly, this command can be overridden at the interface level.

When you want to keep certain routers from exchanging OSPF packets, use the simple form of authentication. The interfaces that the packets are to be sent on still need to be trusted, because the key will be placed in the packets and can be seen by anyone with access to the network.

When you do not trust other users of your network, use MD5 authentication. The system works by using shared secret keys. Because the keys are used to sign the packets with an MD5 checksum, they cannot be forged or tampered with. Because the keys are not included in the packet, snooping the key is not possible. Users of the network can still snoop the contents of packets, however, because the packets are not encrypted.

Advanced Routing Suite’s MD5 authentication is compliant with the specification in OSPF RFC 2328. This specification uses the MD5 algorithm and an authentication key of up to 16 characters. RFC 2328 allows multiple MD5 keys per interface. Each key has two associated time ranges.

Note: In order to turn off authentication, you must include the authentication type in the no form of the command. For example, if MD5 authentication was configured, then simply specifying `no area authentication` will not work. In addition, you must specify the simple key when turning off simple authentication, and you must specify the MD5 `id_number` when turning off MD5 authentication.

Default
The default is for no authentication to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures simple authentication in OSPF instance 1.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 authentication simple abc
(config-router-ospf)# exit
(config)#
```

Example 2
The following example configures md5 authentication for OSPF area 1.2.3.4. The ID for this md5 configuration is 1, and the key is configured as “bar”.

```
(config)# router ospf 2
(config-router-ospf)# area 1.2.3.4 authentication md5 1 bar
(config-router-ospf)# exit
(config)#
```

Example 3
The following example turns off the MD5 authentication that was configured in Example 2. Note that specifying the MD5 key is not required, but specifying the MD5 ID is.
Example 4

The following example configures md5 authentication for OSPF instance 3. The md5 authentication is configured with a start-generate time set to January 02, 2004 at 21:30 hours, a stop-generate time set to January 02, 2004 at 21:45 hours, a start-accept time set to January 02, 2004 at 21:00 hours, and a stop-accept time set to January 02, 2004 at 22:00 hours.

```
(config)# router ospf 3
(config-router-ospf)# area 1.1.1.1 authentication md5 2
          md5 start-generate 2004-01-02.21.30 stop-generate 2004-01-02.21.45
          start-accept 2004-01-02.21.00 stop-accept 2004-01-02.22.00
(config-router-ospf)# exit
(config)#
```

See Also

"authentication" ("authentication" on page 174)

"ip ospf authentication" ("ip ospf authentication" on page 244)

area dead-interval

Name

area dead-interval - specifies the number of seconds that can elapse without receiving a router’s hello packets before the router’s neighbors will declare it down

Syntax

```
area area_id dead-interval time-seconds
no area area_id dead-interval time-seconds?
```

Mode

OSPF Router Configuration

Parameters

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

time-seconds - an integer between 1 and 65535, inclusive specifying an amount of time in seconds

Description

Use the area dead-interval command to specify the amount of time, in seconds, that can elapse without receiving a router’s hello packets before the router’s neighbors will declare it down. This command is specified here at the area level, and it can be overridden in the equivalent interface command. (See "ip ospf dead-interval" ("ip ospf dead-interval" on page 246).)

A general rule for configuring this value is that it should be equal to four times the HELLO interval. (See "area hello-interval" ("area hello-interval" on page 206).) Do not set this value to less than the HELLO interval because convergence will not occur.
The negative form of this command, no area dead-interval, removes the configured time-seconds value and returns this to its default value of 40 seconds. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If area dead-interval is not configured, it is the same as if the user had specified the following:

```
(config-router-ospf)# area area_id dead-interval 40
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a dead-interval of 80 seconds for area 1.2.3.4, and a dead-interval of 60 seconds on interface fxp1.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 dead-interval 80
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf dead-interval 60
(config-if)# exit
(config)#
```

**See Also**

"dead-interval" ("dead-interval" on page 176)

"ip ospf dead-interval" ("ip ospf dead-interval" on page 246)

---

**area filter**

**Name**

area filter  –  filters incoming summary (type 3) LSAs for an area

**Syntax**

```
area area_id filter ipv4_address netmask

no area area_id filter ipv4_address netmask
```

**Mode**

OSPF Router Configuration

**Parameters**

area_id  –  an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

ipv4_address  –  a valid IPv4 filter address specified in dotted-quad format

netmask  –  a valid network mask specified in dotted-quad format

**Description**

The area filter command filters incoming summary (type 3) LSAs for the area. Any incoming summary LSA that falls within the given range will not be advertised into the area. This command has no effect unless the router is an ABR. This command has no effect in Transit areas (i.e. areas containing a Virtual Link). It is
recommended that this command only be used in stub or NSSA areas that have a default being originated into them. Otherwise, addresses falling within the range may not be reachable in the area.

The negative form of this command, no area filter, removes the configured address range from the filter.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example filters summary LSAs in the range 192.168.2.0 255.255.255.0.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 filter 192.168.2.0 255.255.255.0
(config-router-ospf)# exit
(config)#
```

---

**area hello-interval**

**Name**

area hello-interval - specifies the length of time in seconds between Hello packets that the router sends

**Syntax**

area area_id hello-interval time-seconds

no area area_id hello-interval time-seconds

**Mode**

OSPF Router Configuration

**Parameters**

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

time-seconds - an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

**Description**

Use the area hello-interval command to specify the number of seconds between Hello packets sent by this router. This command is specified here at the area level, and it can be overridden in the equivalent interface command. (See "ip ospf hello-interval" ("ip ospf hello-interval" on page 248).)

A general rule for configuring this value is that it should be equal to one-fourth the dead interval. (See "area dead-interval" ("area dead-interval" on page 204).) This value should never be less than the dead interval value.

The negative form of this command, no area hello-interval, removes the configured time-seconds value and returns this to its default value of 10 seconds. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If area hello-interval is not specified, it is the same as if the user had specified the following:
(config-router-ospf)# area area_id hello-interval 10

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a hello interval of 20 seconds for area 1.2.3.4, and a hello interval of 15 seconds on interface fxp1.

    (config)# router ospf 1
    (config-router-ospf)# area 1.2.3.4 hello-interval 20
    (config-router-ospf)# exit
    (config)# interface fxp1
    (config-if)# ip ospf hello-interval 15
    (config-if)# exit
    (config)#

See Also

"area hello-interval" ("area hello-interval" on page 206)
"ip ospf hello-interval" ("ip ospf hello-interval" on page 248)

area nssa

Name

area nssa - configures the area as a Not So Stubby Area (NSSA) according to draft-ietf-ospf-nssa-update-11

Syntax

    area area_id nssa metric metric-type
    no area area_id nssa

Mode

OSPF Router Configuration

Parameters

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

metric - the default OSPF metric. This value can be an integer from 1 to 65535, inclusive. If area nssa is configured, then the default is to not originate a default route into the area.

metric-type - specify 1 or 2 as the default metric type. Routes exported from the Advanced Routing Suite routing table into OSPF default to becoming type 1 ASEs.

Description

The area nssa command configures the specified area as an NSSA according to draft-ietf-ospf-nssa-update-11. If the router is an ABR and has the highest router ID of all the ABRs in the area, and no other ABR in the area is configured to translate always, it will translate Type 7 LSAs with the P-bit (set by the propagate flag in an export command) to Type 5 LSAs. When an ABR that was translating loses a translator election, it will cease translating, and after a number of seconds (determined by the "nssa-stability-interval" ("nssa-stability-interval" on page 186)), it will flush any Type 5 LSAs resulting from translation. Any Type 5 LSAs resulting from direct translation of Type 7 LSAs will be allowed to age out.
If `metric` is configured, then an ABR will originate a default route with that metric into the area. If there are any filters on the area, this will be a type-3 default. Otherwise it will be a type-7 default, with metric type of `metric-type`.

**Note:** NSSA and Stub are mutually exclusive.

**Default**
This command is not explicitly configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example changes the default metric type to 2.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 nssa 10 2
(config-router-ospf)# exit
(config)#
```

**area nssa-range**

**Name**

area nssa-range - specifies the net ranges that should be translated into Type 5 LSAs from NSSA Type 7 LSAs

**Syntax**

```
area area_id nssa-range ip_address mask [no-advertise]?
no area area_id nssa-range ip_address mask [no-advertise]?
```

**Mode**
OSPF Router Configuration

**Parameters**

- `area_id` - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
- `ip_address` - the IPv4 address associated with this range
- `mask` - the network mask for the specified IP address
- `no-advertise` - sets the address range status to DoNotAdvertise. The Type 3 summary LSA will not be suppressed, and the component networks will remain hidden from other networks. Specifying this is optional.

**Description**
The `area nssa-range` command specifies the net ranges that should be translated into Type 5 LSAs from NSSA Type 7 LSAs. The default behavior is to translate Type 7 LSAs that do not fall within a configured net range. This command is valid only in an NSSA. It will be ignored when configured in a non-NSSA.

**Default**
NSSA net ranges are not configured by default.
Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures an NSSA range on address 192.168.110.0 with a mask of 255.255.0.0.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 nssa-range 192.168.110.0 255.255.0.0
(config-router-ospf)# exit
(config)#
```

area nssa-translate-always

Name
area nssa-translate-always - allows an Area Border Router (ABR) to always translate Type 7 LSAs into Type 5 LSAs for the specified area

Syntax
```
area area_id nssa-translate-always
no area area_id nssa-translate-always
```

Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

Description
An NSSA ABR can translate Type 7 LSAs into Type 5 LSAs. Ordinarily, a translator election is run to determine which on ABR for a given NSSA will translate. This option, however, allows an ABR to always translate Type 7 LSAs into Type 5 LSAs for the containing area. If an ABR is configured to always translate, then no other ABR will translate unless it is also configured to always translate.

Default
If area nssa-translate-always is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# no area area_id nssa-translate-always
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures area 1.2.3.4 to always translate Type 7 LSAs into Type 5 LSAs.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 nssa-translate-always
(config-router-ospf)# exit
(config)#
```
area poll-interval

Name

area poll-interval - specifies the length of time, in seconds, between OSPF packets that the router send before adjacency is established with a neighbor

Syntax

area area_id poll-interval time-seconds
no area area_id poll-interval time-seconds?

Mode

OSPF Router Configuration

Parameters

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
time-seconds - the length of time in seconds, specified as an integer between 1 and 65535, inclusive

Description

The area poll-interval command specifies the length of time, in seconds, between OSPF packets that the router sends before adjacency is established with a neighbor. Utilizing this command reduces network overhead in cases where a router may have a neighbor on a given interface at the expense of initial convergence time.

Specifying a poll interval for an area overrides the default configured poll interval. Similarly, an interface-specific poll interval overrides a value configured here.

The negative form of this command, no area poll-interval, removes the configured time-seconds value and returns this to its default value of 120 seconds. Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If area poll-interval is not specified, it is the same as if the user had specified the following:

(config-router-ospf)# area area_id poll-interval 120

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures an area poll interval of 100 seconds. This value is overridden in interface fxp1, where it is configured to be 110 seconds.

(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 poll-interval 100
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf poll-interval 110
(config-if)# exit
(config)#

See Also

"poll-interval" ("poll-interval" on page 186)
area priority

Name
area priority - specifies the priority for becoming the designated router (DR)

Syntax
area area_id priority level
no area area_id priority level?

Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
level - a priority number between 0 and 255, inclusive, for becoming a DR

Description
The area priority command specifies the priority for becoming the DR. The priority specified at the area level can be overridden at the interface level. When more than one router attached to a network attempts to become the DR, the one with the highest priority wins. If the competing routers have the same priority, the one with the highest router ID becomes the DR. The router coming in second in the election becomes the backup DR. A router with a priority set to 0 is ineligible to become the DR.

Note: This command applies only to broadcast or NBMA media.

The negative form of this command, no area priority, removes the configured priority and returns this to its default value of 0. Note: Specifying a value for level in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If the area priority command is not specified, it is the same as if the user had specified the following:

(config-router-ospf)# area area_id priority 1

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures an area priority of 10. It also configures the priority for interface fxp1 to be 5.

(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 priority 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf priority 5
(config-if)# exit
(config)#
See Also
"priority" ("priority" on page 187)
"ip ospf priority" ("ip ospf priority" on page 253)

area range

Name
area range - configures the scope of an area on an Area Border Router (ABR)

Syntax
area area_id range ip_address mask [no-advertise]?
no area area_id range ip_address mask [no-advertise]?

Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

ip_address - the IPv4 address associated with this range

mask - the network mask for the specified IP address

no-advertise - sets the address range status to DoNotAdvertise. The Type 3 summary LSA will not be suppressed, and the component networks will remain hidden from other networks. Specifying this is optional.

Description
The area range command configures the scope of an area on an ABR. Intra-area LSAs that fall within the specified ranges are not advertised into other areas as inter-area routes. Instead, the specified ranges are advertised as summary network LSAs. If no-advertise is specified, the summary network LSAs and all LSAs within the range are not advertised. Intra-area LSAs that do not fall into any range are also advertised as summary network LSAs.

On well-designed networks, the area range command reduces the amount of routing information propagated between areas. The entries in this command can be either networks or subnetwork/mask-length pairs.

Specifying the area range command on a non-ABR will have no effect. The negative form of this command removes the configured summarization.

Default
The area range command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures an area range on address 192.168.0.0 with a mask of 255.255.0.0.

    (config)# router ospf 1
    (config-router-ospf)# area 1.2.3.4 range 192.168.0.0 255.255.0.0
    (config-router-ospf)# exit
    (config)#
area retransmit-interval

**Name**
area retransmit-interval - specifies the number of seconds between link state advertisement (LSA) retransmissions for adjacencies

**Syntax**
area area_id retransmit-interval time-seconds
no area area_id retransmit-interval time-seconds?

**Mode**
OSPF Router Configuration

**Parameters**
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
time-seconds - an integer between 1 and 65535, inclusive, specifying a number of seconds

**Description**
The *area retransmit-interval* command sets the default for the number of seconds between LSA retransmissions for adjacencies. If a Link State Protocol (LSP) is not acknowledged within the amount of time specified in this command, then it is re-sent. This setting is another convergence/network traffic trade-off.

This command overrides the global retransmit interval and can be overridden at the interface level. The negative form of this command, *no area retransmit-interval*, removes the configured value and returns this to its default value of 5 seconds. **Note:** Specifying a value for *time-seconds* in the *no* form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If *area retransmit-interval* is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# area area_id retransmit-interval 5
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the retransmit interval for area 1.2.3.4 to be 10 seconds. This value is then overridden in interface fxp1, where it is configured to be 7 seconds.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 retransmit-interval 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf retransmit-interval 7
(config-if)# exit
(config)#
```

**See Also**
"retransmit-interval" ("retransmit-interval" on page 195)
area stub

Name
area stub - configures an area as a stub area

Syntax
area area_id stub metric?
no area area_id stub metric?

Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
metric - an integer between 1 and 65535, inclusive. If a stub area is configured, then the default is to not originate a default route into the area.

Description
The area stub command configures the specified area as a stub. A stub area is one in which there are no ASE or NSSA routes. Each router in the area must specify that the area is a stub, or adjacencies will not form. If a metric is specified, then it is used to inject a default route into the area with the specified value originating from this router. A metric value should only be specified on an Area Border Router (ABR). It is possible to use stub on multiple ABRs and give them different metrics.

The negative of this command, no area stub, removes the configured stub area.

Note: NSSA and stub are mutually exclusive.

Default
The default type of area is non-stub and non-NSSA. The default for a stub area is to not advertise a type-3 summary. A default is only originated if a cost is given.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures area 4.3.2.1 as a stub area with a metric of 15.

```
(config)# router ospf
(config-router-ospf)# area 4.3.2.1 stub 15
(config-router-ospf)# exit
(config)#
```

area stubhost

Name
area stubhost - specifies directly attached hosts that should be advertised as reachable from the router and the metrics with which they should be advertised
Syntax

```
area area_id stubhost ip_address metric_value
no area area_id stubhost ip_address metric_value
```

Mode

OSPF Router Configuration

Parameters

- **area_id** – an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of `0.0.0.0` signals that the area is a backbone.
- **ip_address** – the address of the host to be advertised
- **metric_value** – the metric to be advertised for the host specified as an integer between 1 and 65535, inclusive. If a stubhost is configured, then this value defaults to 10.

Description

Use the `area stubhost` command to specify directly attached hosts that should be advertised as reachable from the router and the metrics with which they should be advertised. Point-to-point interfaces on which it is not desirable to run OSPF should be specified here. It is also useful to assign an additional address to the loopback interface (one not on the 127 network) and advertise it as a stubhost.

If this address is the same as the router ID, it enables routing to OSPF routers by router ID instead of by interface address. Routing by router ID is more reliable than routing to one of the router’s interface addresses, which may not always be reachable.

The negative of this command, `no area stubhost`, removes the configured stubhost.

**Note:** This command is identical to "area stubnetwork" (“area stubnetwork” on page 215) in function except that a 32-bit mask is assumed.

Default

The `area stubhost` command is not explicitly configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a single stubhost, 192.1.1.1 with a metric of 2 in area 1.2.3.4.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 stubhost 192.1.1.1 2
(config-router-ospf)# exit
(config)#
```

area stubnetwork

Name

**area stubnetwork** – specifies directly attached networks that should be advertised as reachable from this router and the metrics with which they should be advertised

Syntax

```
area area_id stubnetwork ip_address mask metric
no area area_id stubnetwork ip_address mask metric
```
Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

ip_address - the address of the host to be advertised

mask - a mask for a network prefix specified in dotted-quad format

metric - the metric for the stubnetwork specified as an integer between 1 and 65535, inclusive. If a stubnetwork is configured, then this value defaults to 10.

Description
Use the area stubnetwork command to specify directly attached networks that should be advertised as reachable from this router. Interfaces on which it is not desirable to run OSPF should be specified here. No checking is currently done on whether the specified network is actually reachable from this router, so care should be taken.

The negative form of this command, no area stubnetwork, removes the configured stubnetwork.

Note: This command is identical to "area stubhost" ("area stubhost" on page 214) in function except that a 32-bit mask is not assumed and, therefore, must be configured.

Default
The area stubnetwork command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a single stubnetwork, 192.168.10 with a netmask of 24 in area 1.2.3.4.

```
(config)# router ospf 2
(config-router-ospf)# area 1.2.3.4 stubnetwork 192.168.10 255.255.255.0
(config-router-ospf)# exit
(config)#
```

area transmit-delay

Name
area transmit-delay - specifies the number of seconds required to transmit a link state update

Syntax

```
area area_id transmit-delay time-seconds
no area area_id transmit-delay time-seconds?
```

Mode
OSPF Router Configuration

Parameters
area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.
**Description**

The `area transmit-delay` command sets the estimated number of seconds required to transmit a link state update. This value takes into account transmission and propagation delays and must be greater than 0.

This value overrides the global transmit-delay value and can be overridden at the interface level. The negative form of this command, `no area transmit-delay`, removes the configured value and returns this to its default value of 1 second. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `area transmit-delay` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# area area_id transmit-delay 1
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the transmit-delay for area 1.2.3.4 to be 10 seconds. This value is then overridden on interface fxp1, where it is configured to be 5 seconds.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 transmit-delay 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf transmit-delay 5
(config-if)# exit
(config)#
```

**See Also**

"transmit-delay" ("transmit-delay" on page 200)
"ip ospf transmit-delay" ("ip ospf transmit-delay" on page 258)

---

**area virtual-link**

**Name**

area virtual-link - configures a virtual link on the backbone

**Syntax**

```
area transit_area virtual-link neighbor-id
   [ authentication [ (md5 id md5key) | (simple key) ] ]? ||
   [(hello-interval time-seconds) ]? ||
   [(retransmit-interval time-seconds) ]? ||
   [(transmit-delay time-seconds) ]? ||
   [(dead-interval time-seconds) ]?
   no area transit_area virtual-link neighbor-id
   [ authentication [ (md5 id md5key) | (simple key) ] ]? ||
   [(hello-interval time-seconds) ]? ||
   [(retransmit-interval time-seconds) ]? ||
```
Mode

OSPF Router Configuration

Parameters

transit_area – an ID, in dotted-quad format, that sets the area through which the virtual link should exist. This value cannot be 0.0.0.0.

neighbor-id – the ID of one end of the virtual link

[authentication [ (md5 id "md5key") | (simple "key") ]]] – optionally specify an authentication type to include on this virtual link. md5 id md5_key specifies the authentication used for specifying md5 cryptographic authentication. The value for id is an integer between 1 and 255, inclusive. The value for md5_key is a one- to sixteen-character string. simple key specifies simple (clear password) authentication. The value for key is a one- to eight-character string.

hello-interval time-seconds – optionally specify the time in seconds between hello packets that the IOS software sends on an interface. This value must be an unsigned integer and must be the same for all routers and access servers attached to a common network. The default value is 10 seconds.

retransmit-interval time-seconds – optionally specify the time in seconds between link state advertisement (LSA) retransmissions for adjacencies belonging on the interface. This value must be an integer and must be greater than the expected round-trip delay. The default value is 5 seconds.

transmit-delay time-seconds – optionally specify the time in seconds required to send a link state update packet on the interface. This value must be an integer greater than 0. LSAs in the update packet have their age incremented by this amount before transmission. The default value is 1 second.

dead-interval time-seconds – optionally specify the time in seconds that Hello packets are sent before a neighbor declares the router down. The default value is four times the hello-interval, or 40 seconds. If this value is configured less than the hello-interval, convergence will not occur. This value must be the same for all routers and access servers attached to a common network.

Description

Virtual links are used to establish or increase connectivity of the backbone area. The transit_area sets the area through which the virtual link should exist. The neighbor-id represents one end of the virtual link. In addition, all standard interface parameters can optionally be specified on a virtual link.

The negative form of this command, no area virtual-link, removes the configured virtual link. The transit_area and neighbor-id parameters are the minimum required in the negative form of this command to remove the entire virtual link.

Default

The default is for no virtual links to be explicitly configured.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a virtual link in transit area 1.1.1.1 with a neighbor ID of 1.2.2.1, a hello interval of 20 seconds, and a retransmit interval of 20 seconds.

```
(config)# router ospf 1
(config-router-ospf)# area 1.1.1.1 virtual-link 1.2.2.1
hello-interval 20 retransmit-interval 20
(config-router-ospf)# exit
(config)#
```
default-metric

Name
default-metric - specifies the default ASE export metric

Syntax
default-metric metric_value
no default-metric metric_value?

Mode
OSPF Router Configuration

Parameters
metric_value - an integer assigned to exported reachability, specified as a value between 1 and 65535, inclusive

Description
This command is used when exporting a non-OSPF route from the Advanced Routing Suite routing table into OSPF as an ASE route. The default-metric command can be explicitly overridden in export policy.

The negative of this command, no default-metric, removes the configured value and returns this to its default value of 10. **Note:** Specifying a value for metric_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If default-metric is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# default-metric 10
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a default metric of 5 for instance 1.

```
(config)# router ospf 1
(config-router-ospf)# default-metric 5
(config-router-ospf)# exit
(config)#
```

default-nssa-metric

Name
default-nssa-metric - specifies the default cost when exporting non-OSPF routes into OSPF NSSA

Syntax
default-nssa-metric metric_value
no default-nssa-metric metric_value?

Mode
OSPF Router Configuration
**Parameters**

`metric_value` - an integer assigned to exported reachability, specified as a value between 1 and 65535, inclusive

**Description**

default-nssa-metric is used when exporting a non-OSPF route from the Advanced Routing Suite routing table into OSPF as an NSSA route. This command configures the default metric on originated NSSA routes for an OSPF instance. default-nssa-metric can be explicitly overridden in export policy.

The negative form of this command, no default-nssa-metric, removes the configured value and returns the `metric_value` to its default of 1. **Note:** Specifying a value for `metric_value` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If default-nssa-metric is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# default-nssa-metric 10
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the default NSSA metric for OSPF instance 1 to be 4.

```
(config)# router ospf 1
(config-router-ospf)# default-nssa-metric 4
(config-router-ospf)# exit
(config)#
```

default-nssa-type

**Name**

default-nssa-type - changes the default type of routes exported from the Advanced Routing Suite routing table into OSPF NSSA

**Syntax**

```
default-nssa-type [1 | 2]
no default-nssa-type [1 | 2]?
```

**Mode**

OSPF Router Configuration

**Parameters**

`1 | 2` - sets the default metric type to 1 or 2

**Description**

The default-nssa-type command sets the default NSSA metric type to either 1 or 2. Routes exported from the Advanced Routing Suite routing table into OSPF default to becoming type 1 NSSAs. This default can be explicitly change here and overridden in export policy.

The negative form of this command, no default-nssa-type, removes the configured value and returns this to its default value of 1. **Note:** Specifying type 1 or 2 in the no form has no effect on the configuration. Thus, it is displayed above as optional.
Default
If default-nssa-type is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# default-nssa-type 1
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example changes the default NSSA Type to 2.

```
(config)# router ospf 2
(config-router-ospf)# default-nssa-type 2
(config-router-ospf)# exit
(config)#
```

default-preference

Name
default-preference - specifies the way that active routes learned from the OSPF ASE will be selected, compared to other routes

Syntax
```
default-preference pref
do default-preference pref?
```

Mode
OSPF Router Configuration

Parameters

pref - an integer between 1 and 255, inclusive

Description
The default-preference command specifies how active routes that are learned from the OSPF ASE (compared to other routes) will be selected. When a route has been learned from more than one protocol, the active route will be selected from the protocol with the lowest preference. Each protocol has a default preference in this selection.

The negative form of this command, no default-preference, removes the configured preference value and returns this to its default value of 10. Note: Specifying a value for pref in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If default-preference is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# default-preference 10
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the OSPF default preference value to be 100.
(config)# router ospf 2
(config-router-ospf)# default-preference 100
(config-router-ospf)# exit

default-tag

Name
default-tag - used to propagate data from an exterior gateway protocol (such as BGP) through OSPF

Syntax
default-tag tag_value
no default-tag tag_value?

Mode
OSPF Router Configuration

Parameters
tag_value - a integer from 0 to 2147483647, inclusive

Description
OSPF ASE routes have a 32-bit tag field that is not used by the OSPF protocol but can be used when exporting to protocols other than OSPF. This command sets the tag, which can be overridden in export policy.

The negative form of this command, no default-tag, removes the configured tag value and returns this to its default value of 0. Note: Specifying a value for tag_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If default-tag is not specified, it is the same as if the user had specified the following:

(config-router-ospf)# default-tag 0

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the default tag value to be 10.

(config)# router ospf 1
(config-router-ospf)# default-tag 10
(config-router-ospf)# exit
(config)#

default-type

Name
default-type - changes the default type of routes exported from the Advanced Routing Suite routing table into OSPF ASE
Syntax

```
default-type [1 | 2]
no default-type [1 | 2]?
```

**Mode**

OSPF Router Configuration

**Parameters**

- `1 | 2` - sets the default metric type to 1 or 2

**Description**

Routes exported from the Advanced Routing Suite routing table into OSPF default to becoming type 2 ASEs. This default can be explicitly changed here and overridden in export policy.

The negative form of this command, `no default-type`, removes the configured value and returns this to its default value of 1. **Note**: Specifying 1 or 2 as the type in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `default-type` is not specified, it is the same as if the user had specified the following:

```
(config-router-ospf)# default-type 2
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example changes the default metric type to 1.

```
(config)# router ospf 2
(config-router-ospf)# default-type 1
(config-router-ospf)# exit
(config)#
```

---

**advertise-subnet**

**Name**

`advertise-subnet` - specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP

**Syntax**

```
advertise-subnet
no advertise-subnet
```

**Mode**

Area Interface Configuration

**Parameters**

- `none`
Description
The `advertise-subnet` command specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP. Because the netmask is sometimes set improperly on point-to-point interfaces, this option disabled by default.

Default
If `advertise-subnet` is not specified, it is the same as if the user had specified the following:

```
(config-if-ip-ospf)# no advertise-subnet
```

Command History
NGC 2.3 - This command was introduced.

Examples
The following example turns `advertise-subnet` on for area 2.2.2.2 on interface ppp0.

```
(config)# interface ppp0
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# advertise-subnet
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"ip ospf area" ("ip ospf area" on page 243)

allow-all

Name
`allow-all` - permits packets from all routers

Syntax
```
allow-all
no allow-all
```

Mode
Area Interface Configuration

Parameters
none

Description
The `allow-all` command specifies to allow packets from all routers, including those not specified in the "neighbor" ("neighbor" on page 231) command.

Default
If `allow-all` is not specified, it is the same as if the user had specified the following:

```
(config-if-ip-ospf)# no allow-all
```
Command History
NGC 2.3 - This command was introduced.

Examples
The following example specifies that area 1.2.3.4 on interface fxp1 will allow packets from all routers.

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# allow-all
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"ip ospf area" ("ip ospf area" on page 243)

authentication

Name
authentication - specify the type of authentication and key values for an interface

Syntax
```
authentication [ [simple key] | [md5 id_number md5_key
[(start-generate date_time) ||
(stop-generate date_time) || (start-accept date_time)
(stop-accept date_time)]? ] ]
no authentication [ simple key | md5 id_number ]
```

Mode
Area Interface Configuration

Parameters
simple key - specifies simple (clear password) authentication. The value for key is specified as a one-to eight-character string.

md5 id_number md5_key [(start-generate date_time) ||
(stop-generate date_time) || (start-accept date_time)
(stop-accept date_time)]? ] - specifies the authentication used for specifying md5 cryptographic authentication. The value for id_number is an integer with a value between 1 and 255, inclusive. The value for md5_key is a one- to sixteen-character string. The start and stop values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.

Description
Authentication can help to guarantee that routing information is imported only from trusted routers. A variety of authentication schemes can be used, but a single scheme must be configured for each network. The use of different schemes enables some interfaces to use much stricter authentication than others. The two authentication schemes available are simple, and MD5.

The authentication command specifies the type of authentication and key values used for an area on an interface. This is used by OSPF authentication to generate and verify the authentication field in the OSPF header. If configured, the area authentication ("area authentication" ("area authentication" on page 202)) is the default unless it is configured here at the area interface level.

When you want to keep certain routers from exchanging OSPF packets, use the simple form of authentication. The interfaces that the packets are to be sent on still need to be trusted, because the key will be placed in the packets and can be seen by anyone with access to the network.
When you do not trust other users of your network, use MD5 authentication. The system works by using shared secret keys. Because the keys are used to sign the packets with an MD5 checksum, they cannot be forged or tampered with. Because the keys are not included in the packet, snooping the key is not possible. Users of the network can still snoop the contents of packets, however, because the packets are not encrypted.

Advanced Routing Suite’s MD5 authentication is compliant with the specification in OSPF RFC 2328. This specification uses the MD5 algorithm and an authentication key of up to 16 characters. RFC 2328 allows multiple MD5 keys per interface. Each key has two associated time ranges.

**Note:** In order to turn off authentication, you must include the authentication type in the `no` form of the command. For example, if MD5 authentication was configured, then simply specifying `no authentication` will not work. In addition, you must specify the simple `key` when turning off simple authentication, and you must specify the MD5 `id_number` when turning off MD5 authentication.

**Default**

Authentication is not explicitly configured by default.

**Command History**

NGC 2.3 - This command was introduced.

**Examples**

**Example 1**

The following example configures the global authentication to be the simple keyword "abc". This authentication is then overridden for area 1.2.3.4 on interface fxp0 to be "bar".

```
(config)# router ospf 1
(config-router-ospf)# authentication simple abc
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# authentication simple bar
(config-if-ip-ospf)# exit
(config-if)#
```

**Example 2**

The following example configures md5 authentication for area 2.2.2.2 on interface fxp1. The ID for this md5 configuration is 1, and the key is configured as "bar".

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# authentication md5 1 bar
(config-if-ip-ospf)# exit
(config-if)#
```

**Example 3**

The following example turns off the MD5 authentication that was configured in Example 2.

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# no authentication md5 1
(config-if-ip-ospf)# exit
(config-if)#
```
Example 4
The following example configures md5 authentication for area 4.3.2.1 on interface fxp3. The md5 authentication is configured with a start-generate time set to January 02, 2004 at 21:30 hours, a stop-generate time set to January 02, 2004 at 21:45 hours, a start-accept time set to January 02, 2004 at 21:00 hours, and a stop-accept time set to January 02, 2004 at 22:00 hours.

```
(config)# interface fxp3
(config-if)# ip ospf 1 area 4.3.2.1
(config-if-ip-ospf)# authentication md5 2 md5 start-generate 2004-01-02.21.30 stop-generate 2004-01-02.21.45 start-accept 2004-01-02.21.00 stop-accept 2004-01-02.22.00
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"authentication" ("authentication" on page 174)
"area authentication" ("area authentication" on page 202)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf authentication" ("ip ospf authentication" on page 244)

cost

Name
cost - specifies the cost for a route to transmit an interface

Syntax
```
cost cost_value
no cost cost_value?
```

Mode
Area Interface Configuration

Parameters
cost_value - the cost to be associated with this interface, specified as an integer between 1 and 65535

Description
The cost command is used for specifying the cost for a route to transmit an interface. This command can be explicitly overridden in export policy. The negative form of this command, no cost, removes the configured cost and returns this to its default value of 1. Note: Specifying a value for cost_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If cost is not specified, it is the same as if the user had specified the following:
```
(config-if-ip-ospf)# cost 10
```

Command History
NGC 2.3 - This command was introduced.
Examples
The following example configures the cost for area 1.2.3.4 on interface fxp1 to be 15.

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# cost 15
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"ip ospf area" ("ip ospf area" on page 243)

dead-interval

Name
dead-interval - specifies the number of seconds that may elapse on this interface without receiving a router’s hello packets before the router’s neighbors will declare it down

Syntax
```
dead-interval time.seconds
no dead-interval time.seconds?
```

Mode
Area Interface Configuration

Parameters
time.seconds - an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

Description
Use the dead-interval command to specify the amount of time, in seconds, that can elapse without receiving a router’s hello packets before the router’s neighbors will declare it down. This command is specified here at the interface level. It overrides a dead-interval configured at the area and global levels.

A general rule for configuring this value is that it should be equal to four times the HELLO interval. (See "hello-interval" ("hello-interval" on page 180).) Do not set this value to less than the HELLO interval because convergence will not occur.

The negative form of this command, no dead-interval, removes the configured time.seconds value and returns this to its default value of 40 seconds. Note: Specifying a value for time.seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If dead-interval is not specified, it defaults to the value specified in "area dead-interval" ("area dead-interval" on page 204) then the global "dead-interval" ("dead-interval" on page 176). If neither of these is specified, however, then it is the same as if the user had specified the following:

```
(config-if-ip-ospf)# dead-interval 40
```

Command History
NGC 2.3 - This command was introduced.
Examples
The following example configures a global dead-interval of 80 seconds and a dead-interval of 60 seconds for area 1.2.3.4 on interface fxp1.

```text
(config)# router ospf 1
(config-router-ospf)# dead-interval 80
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# dead-interval 60
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"dead-interval" ("dead-interval" on page 176)
"area dead-interval" ("area dead-interval" on page 204)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf dead-interval" ("ip ospf dead-interval" on page 246)

cenable

Name
cenable - enables OSPF on an interface
c
Syntax
c  enable
c  no enable

cMode
cArea Interface Configuration

cParameters
cnone

cDescription
cThe enable command specifies to run OSPF on an interface.
c
Default
cBy default, OSPF runs on configured interfaces. Therefore, if enable is not specified, it is the same as if the user had specified the following:

```text
(config-if-ip-ospf)# enable
```

cCommand History
cNGC 2.3 - This command was introduced.
c
cExamples
cThe following example disables interface fxp1 for area 1.2.3.4.

```text
(config)# interface fxp1
```
Hello-interval

Name

Hello-interval - specifies the length of time in seconds between hello packets that the router sends on the interface for the specified area.

Syntax

Hello-interval time-seconds

Mode

Area Interface Configuration

Parameters

time-seconds - an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

Description

Use the hello-interval command to specify the number of seconds between Hello packets sent by the router on this interface for the specified area. This command overrides the interval configured in the equivalent global and area commands. (See "hello-interval" ("hello-interval" on page 180) and "area hello-interval" ("area hello-interval" on page 206).)

A general rule for configuring this value is that it should be equal to one-fourth the dead interval. (See "dead-interval" ("dead-interval" on page 228).) This value should never be less than the dead interval value.

The negative form of this command, no hello-interval, removes the configured time-seconds value.

Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If hello-interval is not specified, it defaults to the value specified in "area hello-interval" ("area hello-interval" on page 206) then in the global "hello-interval" ("hello-interval" on page 180). If neither the area nor global hello interval is specified, however, then it is the same as if the user had specified the following:

(config-if-ip-ospf)# hello-interval 10

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures a global hello interval of 20 seconds and a hello interval of 15 seconds for area 1.2.3.4 on interface fxp1.

(config)# router ospf 1

(config-router-ospf)# hello-interval 20
neighbor

Name
neighbor - specifies neighboring routes, and specifies whether those routes are eligible to become the designated router (DR)

Syntax
neighbor ip_address [eligible]?
no neighbor ip_address [eligible]?

Mode
Area Interface Configuration

Parameters
ip_address - a valid IPv4 address specified in dotted-quad notation
eligible - optionally specify whether the indicated address is eligible to become the DR

Description
By definition it is not possible to send broadcast or multicast packets to discover OSPF neighbors on a non-broadcast medium, so all neighbors must be configured. The host address list includes one or more neighbors. The eligible option is used to indicate an address’s eligibility to become a DR. Refer to section 9.5.1 of RFC 2328 for more information.

The negative form of this command, no neighbor, removes the configured neighbors.

Default
OSPF neighbors are not explicitly configured by default.

Command History
NGC 2.3 - This command was introduced.

Examples
The following example specifies that host 10.1.1.1 is eligible to become a DR in area 2.2.2.2.

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# neighbor 10.1.1.1 eligible
(config-if-ip-ospf)# exit
```
network

Name
	network - specifies a either a nonbroadcast interface on an NBMA medium or a point-to-multipoint interface

Syntax

network [ nonbroadcast | point-to-multipoint ]

no network [ nonbroadcast | point-to-multipoint ]

Mode

Area Interface Configuration

Parameters

nonbroadcast - specifies a nonbroadcast interface on an NBMA medium

point-to-multipoint - specifies a point-to-multipoint interface

Description

Use the network command to specify either a nonbroadcast or point-to-point interface. Because an OSPF broadcast medium must support IP multicasting, a broadcast-capable medium that does not support IP multicasting must be configured as a nonbroadcast interface. This includes the loopback interface on many operating systems.

Default

The default mode is broadcast/point-to-point; therefore this command is not explicitly configured by default.

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures interface lo1 as a nonbroadcast interface in area 2.2.2.2.

```
(config)# interface lo1
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# network nonbroadcast
(config-if-ip-ospf)# exit
(config-if)#
```

See Also

"ip ospf area" ("ip ospf area" on page 243)

no-multicast

Name

no-multicast - disables multicast on a specified interface
Syntax

no-multicast
no no-multicast

Mode
Area Interface Configuration

Parameters
none

Description
The no-multicast command explicitly disables multicast for an area on a specified interface. The negative form of this command, no no-multicast, re-enables multicast on the area interface.

Default
By default, interfaces are multicast interfaces. Therefore, if no-multicast is not specified, it is the same as if the user had specified the following:

(config-if-ip-ospf)# no no-multicast

Command History
NGC 2.3 - This command was introduced.

Examples
The following example disables multicast for area 2.2.2.2 on interface fxp2.

(config)# interface fxp2
(config-if)# ip ospf 1 area 2.2.2.2
(config-if-ip-ospf)# no-multicast
(config-if-ip-ospf)# exit
(config-if)#

See Also
"ip ospf area" ("ip ospf area" on page 243)

passive-interface

Name
passive-interface - disables reception and transmission on an interface for a specific area

Syntax

passive-interface
no passive-interface

Mode
Area Interface Configuration

Parameters
none
Description
This command specifies that Advanced Routing Suite will neither send nor receive packets on this interface. This is used, for example, when this is the only router on the network. This has the effect of originating a stub link to his interface into the domain.

Note: OSPF passive interface is not used to learn other routers' announcements, which is the way passive works in RIP. If your host is connected to a single network on which there are multiple routers, use Router Discovery combined with ICMP redirects to learn a default route and the best route. If your host is connected directly to multiple networks, this method might produce the best routes.

The negative form of this command, no passive-interface, re-enables reception and transmission for an area on the interface.

Default
If passive-interface is not specified, it is the same as if the user had specified the following:

```
(config-if-ip-ospf)# no passive-interface
```

Command History
NGC 2.3 - This command was introduced.

Examples
The following example specifies to not send or receive packets on area 1.2.3.4 for interface fxp2.

```
(config)# interface fxp2
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# passive-interface
(config-if-ip-ospf)# exit
(config-if)#
```

See Also
"ip ospf area" ("ip ospf area" on page 243)

poll-interval

Name
poll-interval - specifies the length of time, in seconds, between OSPF packets that the router send before adjacency is established with a neighbor

Syntax
```
poll-interval time-seconds
no poll-interval time-seconds?
```

Mode
Area Interface Configuration

Parameters
```
time-seconds - the length of time in seconds, specified as an integer between 1 and 65535, inclusive
```

Description
The poll-interval command specifies the length of time, in seconds, between OSPF packets that the router sends before adjacency is established with a neighbor. Utilizing this command reduces network
overhead in cases where a router may have a neighbor on a given interface at the expense of initial convergence time.

The negative form of this command, no poll-interval, removes the configured time-seconds value and returns this to its default value of 120 seconds. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If poll-interval is not specified, it is the same as if the user had specified the following:

```
(config-if-ip-ospf)# poll-interval 120
```

**Command History**

NGC 2.3 - This command was introduced.

**Examples**

The following configures the poll-interval for area 1.2.3.4 on interface fxp0 to be 100 seconds.

```
(config)# interface fxp0
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# poll-interval 100
(config-if-ip-ospf)# exit
(config-if)#
```

**See Also**

"area poll-interval" ("area poll-interval" on page 210)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf poll-interval" ("ip ospf poll-interval" on page 252)
"poll-interval" ("poll-interval" on page 186)

---

**priority**

**Name**

priority - configures the area-specific priority on an interface for becoming the designated router (DR)

**Syntax**

```
priority level
no priority level?
```

**Mode**

Area Interface Configuration

**Parameters**

level - a priority number between 0 and 255, inclusive, for becoming a DR

**Description**

The priority command configures the area-specific priority on this interface for becoming the DR. This priority overrides any that is configured at the global or area levels. (See "priority" ("priority" on page 187) and "area priority" ("area priority" on page 211).) When more than one router attached to a network attempts to become the DR, the one with the highest priority wins. If the competing routers have the same priority, the one with the highest router ID becomes the DR. The router coming in second in the election becomes the backup DR. A router with a priority set to 0 is ineligible to become the DR.
The negative form of this command, no priority, removes the configured priority. **Note:** Specifying a value for level in the no form has no effect on the configuration. Thus, it is displayed above as optional.  

**Default**  
If the `priority` command is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if-ip-ospf)# priority 1
```

**Command History**  
NGC 2.3 - This command was introduced.

**Examples**  
The following example configures a global priority of 10. It also configures the priority for area 1.2.3.4 on interface fxp1 to be 5.

```
(config)# router ospf 1
(config-router-ospf)# priority 10
(config-router-ospf)# exit
(config)# interface fxp1
(config)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# priority 5
(config-if-ip-ospf)# exit
(config-if)#
```

**See Also**  
"area priority" ("area priority" on page 211)  
"ip ospf area" ("ip ospf area" on page 243)  
"ip ospf priority" ("ip ospf priority" on page 253)  
"priority" ("priority" on page 187)

---

**retransmit-interval**

**Name**

`retransmit-interval` - specifies the number of seconds between link state advertisement (LSA) retransmissions for adjacencies

**Syntax**

```
retransmit-interval time-seconds

no retransmit-interval time-seconds?
```

**Mode**

Area Interface Configuration

**Parameters**

`time-seconds` - the length of time in seconds, between 1 and 65535, inclusive specified as an integer
Description

The `retransmit-interval` command sets the default for the number of seconds between LSA retransmissions for adjacencies. If a Link State Protocol (LSP) is not acknowledged within the number of seconds specified here, it is re-sent.

This command overrides any configured in the global or area levels. (See "retransmit-interval" ("retransmit-interval" on page 195) and "area retransmit-interval" ("area retransmit-interval" on page 213).)

The negative form of this command, `no retransmit-interval`, removes the configured retransmit value. Note: Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If the `retransmit-interval` command is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if-ip-ospf)# retransmit-interval 5
```

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures the global retransmit interval f10 seconds. This value is then overridden in interface fxp1 for the area 1.2.3.4, where it is configured to be 7 seconds.

```
(config)# router ospf 1
(config-router-ospf)# retransmit-interval 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if-ip-ospf)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# retransmit-interval 7
(config-if-ip-ospf)# exit
(config-if)#
```

See Also

"area retransmit-interval" ("area retransmit-interval" on page 213)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf retransmit-interval" ("ip ospf retransmit-interval" on page 254)
"retransmit-interval" ("retransmit-interval" on page 195)

`traffic-eng administrative-weight`

Name

`traffic-eng administrative-weight` - sets the area-specific cost of the interface for Traffic Engineering purposes

Syntax

```
traffic-eng administrative-weight number
no traffic-eng administrative-weight number?
```

Mode

Area Interface Configuration
Parameters

*number* - an integer from 1 to 65535, inclusive

Description

Use the `traffic-eng administrative-weight` command to configure the cost of the interface for Traffic Engineering purposes. This cost can be different than the normal OSPF interface cost. The negative form of this command, `no traffic-eng administrative-weight`, removes the configured `number` value.

Notes:

- Specifying a value for `number` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).

Default

This command is not explicitly configured by default.

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures a traffic engineering cost of 6 for area 1.2.3.4 on interface fxp2.

```
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# traffic-eng administrative-weight 6
(config-if-ip-ospf)# exit
(config)#
```

See Also

"enable-te" ("enable-te" on page 179)
"ip ospf area" ("ip ospf area" on page 243)

**traffic-eng attribute-flags**

Name

`traffic-eng attribute-flags` - sets the OSPF Traffic Engineering attributes for the interface

Syntax

```
traffic-eng attribute-flags number
no traffic-eng attribute-flags number?
```

Mode

Area Interface Configuration
Parameters

number – an integer from 0 to 4,294,967,295, inclusive

Description

Use the traffic-eng attribute-flags command to sets the area-specific OSPF Traffic Engineering attributes for the interface. These flags can be used to indicate which link groups (or "colors") are present on the interface. The negative form of this command, no traffic-eng attribute-flags, removes the configured number value.

Notes:

- Specifying a value for number in the no form has no effect on the configuration. Thus, it is displayed above as optional.
- Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).

Default

This command is not explicitly configured by default.

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures a traffic engineering attribute flag of 1 for area 1.2.3.4 on interface fxp2.

```
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# traffic-eng attribute-flags 1
(config-if-ip-ospf)# exit
(config-if)#
```

See Also

"enable-te" ("enable-te" on page 179)
"ip ospf area" ("ip ospf area" on page 243)

traffic-eng bandwidth

Name

traffic-eng bandwidth – sets the bandwidth of the interface

Syntax

```
traffic-eng bandwidth kbps
no traffic-eng bandwidth kbps
```

Mode

Area Interface Configuration
Parameters

\textit{kbps} - an integer from 0 to 65535, inclusive, specifying a number of kilobytes per second

Description

The \texttt{traffic-eng bandwidth} command configures an area-specific bandwidth of the interface for Traffic Engineering information distributed within the OSPF domain. The negative form of this command, \texttt{no traffic-eng bandwidth}, removes the configured \textit{kbps} value.

Notes:

\begin{itemize}
  \item Specifying a value for \textit{kbps} in the \texttt{no} form has no effect on the configuration. Thus, it is displayed above as optional.
  \item Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).
\end{itemize}

Default

This command is not explicitly configured by default.

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures a traffic engineering bandwidth of 1000 kbps for area 1.2.3.4 on interface fxp2.

\begin{verbatim}
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# traffic-eng bandwidth 1000
(config-if-ip-ospf)# exit
(config-if)#
\end{verbatim}

See Also

"enable-te" ("enable-te" on page 179)
"ip ospf area" ("ip ospf area" on page 243)

\textbf{transmit-delay}

Name

\textit{transmit-delay} - specifies the number of seconds required to transmit a link state update

Syntax

\begin{verbatim}
transmit-delay time-seconds
no transmit-delay time-seconds?
\end{verbatim}

Mode

Area Interface Configuration
Parameters

time-seconds - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

Description

The `transmit-delay` command sets the estimated number of seconds required to transmit a link state update. This command takes into account transmission and propagation delays and must be greater than 0. This command overrides any given at the global or area levels ("transmit-delay" ("transmit-delay" on page 200) and "area transmit-delay" ("area transmit-delay" on page 216)).

The negative form of this command, no `transmit-delay`, removes the configured `time-seconds` value. **Note:** Specifying a value for `time-seconds` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `transmit-delay` is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if-ip-ospf)# transmit-delay 1
```

Command History

NGC 2.3 - This command was introduced.

Examples

The following example configures the global `transmit-delay` for to be 10 seconds. This value is then overridden on interface fxp1 for area 1.2.3.4, where it is configured to be 5 seconds.

```
(config)# router ospf 1
(config-router-ospf)# transmit-delay 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)# transmit-delay 5
(config-if-ip-ospf)# exit
(config-if)#
```

See Also

"area transmit-delay" ("area transmit-delay" on page 216)
"ip ospf area" ("ip ospf area" on page 243)
"ip ospf transmit-delay" ("ip ospf transmit-delay" on page 258)
"transmit-delay" ("transmit-delay" on page 200)

**ip ospf advertise-subnet**

Name

`ip ospf advertise-subnet` - specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP

Syntax

```
ip ospf advertise-subnet
no ip ospf advertise-subnet
```
Mode
Interface Configuration

Parameters
none

Description
The `ip ospf advertise-subnet` command specifies whether OSPF will, when advertising point-to-point interfaces, advertise the network number and netmask of the point-to-point interface instead of a host route to the remote IP. Because the netmask is sometimes set improperly on point-to-point interfaces, this option disabled by default.

Default
If `ip ospf advertise-subnet` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip ospf advertise-subnet
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example turns `advertise-subnet` on for interface ppp0.

```
(config)# interface ppp0
(config-if)# ip ospf advertise-subnet
(config-if)# exit
(config)#
```

`ip ospf allow-all`

Name
`ip ospf allow-all` - permits packets from all routers

Syntax
```
ip ospf allow-all
no ip ospf allow-all
```

Mode
Interface Configuration

Parameters
none

Description
The `ip ospf allow-all` command specifies to allow packets from all routers, including those not specified in the "ip ospf neighbor" ("ip ospf neighbor" on page 249) command.

Default
If `ip ospf allow-all` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip ospf allow-all
```
Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that interface fxp1 will allow packets from all routers.

```
(config)# interface fxp1
(config-if)# ip ospf allow-all
(config-if)# exit
(config)#
```

ip ospf area

Name
ip ospf area - configures an area ID for an interface

Syntax
```
ip ospf instance_id area area_id
no ip ospf instance_id area area_id
```

Mode
Interface Configuration

Parameters
instance_id - specifies to return information for a specific instance. This can be an integer from 1 to 4,294,967,294, inclusive.

area_id - an ID for an area specified as either a decimal value or as an IP address in dotted-quad format. A value of 0.0.0.0 signals that the area is a backbone.

Description
The ip ospf area command sets the OSPF area ID for an interface and enters the user into OSPF Area Interface Configuration mode. As a result, all remaining interface attributes can be configured for this area.

Default
This command is not explicitly configured by default.

Command History
NGC 2.3 - This command was introduced.

Examples
In the following example, area 1.2.3.4 is configured for interface fxp1.

```
(config)# interface fxp1
(config-if)# ip ospf 1 area 1.2.3.4
(config-if-ip-ospf)#
```
ip ospf authentication

**Name**

**ip ospf authentication** - specify the type of authentication and key values for an interface

**Syntax**

```
ip ospf authentication [ [simple key] | [ md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time) || (start-accept date_time) (stop-accept date_time)]? ] ]
no ip ospf authentication [ simple key | md5 id_number ]
```

**Mode**

Interface Configuration

**Parameters**

- **simple key** - specifies simple (clear password) authentication. The value for key is specified as a one- to eight-character string.
- **md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time) || (start-accept date_time) || (stop-accept date_time)]** - specifies the authentication used for specifying md5 cryptographic authentication. The value for id_number is an integer with a value between 1 and 255, inclusive. The value for md5_key is a one- to sixteen-character string. The start and stop values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.

**Description**

Authentication can help to guarantee that routing information is imported only from trusted routers. A variety of authentication schemes can be used, but a single scheme must be configured for each network. The use of different schemes enables some interfaces to use much stricter authentication than others. The two authentication schemes available are simple, and MD5.

The **ip ospf authentication** command specifies the type of authentication and key values used on an interface. This is used by OSPF authentication to generate and verify the authentication field in the OSPF header. If configured, the area authentication ("area authentication" ("area authentication" on page 202)) is the default unless it is configured here at the interface level.

When you want to keep certain routers from exchanging OSPF packets, use the simple form of authentication. The interfaces that the packets are to be sent on still need to be trusted, because the key will be placed in the packets and can be seen by anyone with access to the network.

When you do not trust other users of your network, use MD5 authentication. The system works by using shared secret keys. Because the keys are used to sign the packets with an MD5 checksum, they cannot be forged or tampered with. Because the keys are not included in the packet, snooping the key is not possible. Users of the network can still snoop the contents of packets, however, because the packets are not encrypted.

Advanced Routing Suite’s MD5 authentication is compliant with the specification in OSPF RFC 2328. This specification uses the MD5 algorithm and an authentication key of up to 16 characters. RFC 2328 allows multiple MD5 keys per interface. Each key has two associated time ranges.

**Note:** In order to turn off authentication, you must include the authentication type in the no form of the command. For example, if MD5 authentication was configured, then simply specifying **no ip ospf authentication** will not work. In addition, you must specify the simple key when turning off simple authentication, and you must specify the MD5 id_number when turning off MD5 authentication.

**Default**

Authentication is not explicitly configured by default.
Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures the area authentication to be the simple keyword "abc". This authentication is then overridden for interface fxp0 to be "bar".

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 authentication simple abc
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf authentication simple bar
(config-if)# exit
(config)#
```

Example 2
The following example configures md5 authentication for interface fxp1. The ID for this md5 configuration is 1, and the key is configured as "bar".

```
(config)# interface fxp1
(config-if)# ip ospf authentication md5 1 bar
(config-if)# exit
(config)#
```

Example 3
The following example turns off the MD5 authentication that was configured in Example 2. Note that specifying the MD5 key is not required, but specifying the MD5 ID is.

```
(config)# interface fxp1
(config-if)# no ip ospf authentication md5 1
(config-if)# exit
(config)#
```

Example 4
The following example configures md5 authentication for interface fxp3. The md5 authentication is configured with a start-generate time set to January 02, 2004 at 21:30 hours, a stop-generate time set to January 02, 2004 at 21:45 hours, a start-accept time set to January 02, 2004 at 21:00 hours, and a stop-accept time set to January 02, 2004 at 22:00 hours.

```
(config)# interface fxp3
(config-if)# ip ospf authentication md5 2 md5 start-generate 2004-01-02.21.30 stop-generate 2004-01-02.21.45 start-accept 2004-01-02.21.00 stop-accept 2004-01-02.22.00
(config-if)# exit
(config)#
```

See Also
"authentication" ("authentication" on page 174)
"area authentication" ("area authentication" on page 202)
ip ospf cost

Name

ip ospf cost - specifies the cost for a route to transmit an interface

Syntax

ip ospf cost cost_value
no ip ospf cost cost_value?

Mode

Interface Configuration

Parameters

cost_value - the cost to be associated with this interface, specified as an integer between 1 and 65535

Description

The `ip ospf cost` command is used for specifying the cost for a route to transmit an interface. This command can be explicitly overridden in export policy. The negative form of this command, `no ip ospf cost`, removes the configured cost and returns this to its default value of 1. Note: Specifying a value for `cost_value` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip ospf cost` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip ospf cost 10
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the cost for interface fxp1 to be 15.

```
(config)# interface fxp1
(config-if)# ip ospf cost 15
(config-if)# exit
(config)#
```

ip ospf dead-interval

Name

ip ospf dead-interval - specifies the number of seconds that may elapse on this interface without receiving a router's hello packets before the router's neighbors will declare it down

Syntax

ip ospf dead-interval time-seconds
no ip ospf dead-interval time-seconds?

Mode

Interface Configuration
Parameters

time-seconds – an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

Description

Use the `ip ospf dead-interval` command to specify the amount of time, in seconds, that can elapse without receiving a router’s hello packets before the router’s neighbors will declare it down. This command is specified here at the interface level. It overrides a dead-interval configured at the area and global levels.

A general rule for configuring this value is that it should be equal to four times the HELLO interval. (See "ip ospf hello-interval" ("ip ospf hello-interval" on page 248).) Do not set this value to less than the HELLO interval because convergence will not occur.

The negative form of this command, `no ip ospf dead-interval`, removes the configured `time-seconds` value and returns this to its default value of 40 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip ospf dead-interval` is not specified, it defaults to the value specified in "area dead-interval" ("area dead-interval" on page 204) then the global "dead-interval" ("dead-interval" on page 176). If neither of these is specified, however, then it is the same as if the user had specified the following:

```
(config-if)# ip ospf dead-interval 40
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a dead-interval of 80 seconds in area 1.2.3.4, and a dead-interval of 60 seconds on interface fxp1.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 dead-interval 80
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf dead-interval 60
(config-if)# exit
(config)#
```

See Also

"dead-interval" ("dead-interval" on page 176)
"area dead-interval" ("area dead-interval" on page 204)

---

**ip ospf enable**

**Name**

`ip ospf enable` - enables OSPF on an interface

**Syntax**

```
ip ospf enable
no ip ospf enable
```
Mode
Interface Configuration

Parameters
none

Description
The `ip ospf enable` command specifies to run OSPF on an interface.

Default
By default, OSPF runs on configured interfaces. Therefore, if `ip ospf enable` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip ospf enable
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example disables interface fxp1.
```
(config)# interface fxp1
(config-if)# no ip ospf enable
(config-if)# exit
(config)#
```

`ip ospf hello-interval`

Name
`ip ospf hello-interval` - specifies the length of time in seconds between hello packets that the router sends on the interface

Syntax
```
ip ospf hello-interval time-seconds
no ip ospf hello-interval time-seconds?
```

Mode
Interface Configuration

Parameters
`time-seconds` - an integer between 1 and 65535, inclusive, specifying an amount of time in seconds

Description
Use the `ip ospf hello-interval` command to specify the number of seconds between Hello packets sent by the router on this interface. This command overrides the interval configured in the equivalent global and area commands. (See "hello-interval" ("hello-interval" on page 180) and "area hello-interval" ("area hello-interval" on page 206).)

A general rule for configuring this value is that it should be equal to one-fourth the dead interval. (See "ip ospf dead-interval" ("ip ospf dead-interval" on page 246).) This value should never be less than the dead interval value.
The negative form of this command, `no ip ospf hello-interval`, removes the configured `time-seconds` value. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip ospf hello-interval` is not specified, it defaults to the value specified in "area hello-interval" ("area hello-interval" on page 206) then in the global "hello-interval" ("hello-interval" on page 180). If neither the area nor global hello interval is specified, however, then it is the same as if the user had specified the following:

```
(config-if)# ip ospf hello-interval 10
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a hello interval of 20 seconds in area 1.2.3.4, and a hello interval of 15 seconds on interface fxp1.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 hello-interval 20
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf hello-interval 15
(config-if)# exit
(config)#
```

**See Also**

"hello-interval" ("hello-interval" on page 180)
"area hello-interval" ("area hello-interval" on page 206)

---

**ip ospf neighbor**

**Name**

`ip ospf neighbor` - specifies neighboring routes, and specifies whether those routes are eligible to become the designated router (DR)

**Syntax**

```
ip ospf neighbor ip_address [eligible]?
no ip ospf neighbor ip_address [eligible]?
```

**Mode**

Interface Configuration

**Parameters**

- `ip_address` - a valid IPv4 address specified in dotted-quad notation
- `eligible` - optionally specify whether the indicated address is eligible to become the DR

**Description**

By definition it is not possible to send broadcast or multicast packets to discover OSPF neighbors on a non-broadcast medium, so all neighbors must be configured. The host address list includes one or more
neighbors. The eligible option is used to indicate an address’s eligibility to become a DR. Refer to section 9.5.1 of RFC 2328 for more information.

The negative form of this command, no ip ospf neighbor, removes the configured neighbors.

Default
OSPF neighbors are not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies that host 10.1.1.1 is eligible to become a DR.

```
(config)# interface fxp1
(config-if)# ip ospf neighbor 10.1.1.1 eligible
(config-if)# exit
(config)#
```

**ip ospf network**

**Name**
ip ospf network - specifies a either a nonbroadcast interface on an NBMA medium or a point-to-multipoint interface

**Syntax**
```
ip ospf network [ point-to-multipoint | nonbroadcast ]
no ip ospf network [ point-to-multipoint | nonbroadcast ]
```

**Mode**
Interface Configuration

**Parameters**
nonbroadcast - specifies a nonbroadcast interface on an NBMA medium
point-to-multipoint - specifies a point-to-multipoint interface

**Description**
Use the ip ospf network command to specify either a nonbroadcast or point-to-point interface. Because an OSPF broadcast medium must support IP multicasting, a broadcast-capable medium that does not support IP multicasting must be configured as a nonbroadcast interface. This includes the loopback interface on many operating systems.

**Default**
The default mode is broadcast/point-to-point; therefore this command is not explicitly configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures interface lo1 as a nonbroadcast interface.

```
(config)# interface lo1
```
ip ospf no-multicast

Name
ip ospf no-multicast - disables multicast on a specified interface

Syntax
ip ospf no-multicast
no ip ospf no-multicast

Mode
Interface Configuration

Parameters
none

Description
The ip ospf no-multicast command explicitly disables multicast on a specified interface. The negative form of this command, no ip ospf no-multicast, re-enables multicast on an interface.

Default
By default, interfaces are multicast interfaces. Therefore, if ip ospf no-multicast is not specified, it is the same as if the user had specified the following:

(config-if)# no ip ospf no-multicast

Command History
NGC 2.2 - This command was introduced.

Examples
The following example disables multicast on interface fxp2.

(config)# interface fxp2
(config-if)# ip ospf no-multicast
(config-if)# exit
(config)#

ip ospf passive-interface

Name
ip ospf passive-interface - disables reception and transmission on an interface

Syntax
ip ospf passive-interface
no ip ospf passive-interface
Mode
Interface Configuration

Parameters
none

Description
This command specifies that Advanced Routing Suite will neither send nor receive packets on this interface. This is used, for example, when this is the only router on the network. This has the effect of originating a stub link to his interface into the domain.

Note: OSPF passive interface is not used to learn other routers’ announcements, which is the way passive works in RIP. If your host is connected to a single network on which there are multiple routers, use Router Discovery combined with ICMP redirects to learn a default route and the best route. If your host is connected directly to multiple networks, this method might produce the best routes.

The negative form of this command, no ip ospf passive-interface, re-enables reception and transmission on the interface.

Default
If ip ospf passive-interface is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip ospf passive-interface
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies to not send or receive packets on interface fxp2.

```
(config)# interface fxp2
(config-if)# ip ospf passive-interface
(config-if)# exit
(config)#
```

**ip ospf poll-interval**

Name
ip ospf poll-interval – specifies the length of time, in seconds, between OSPF packets that the router send before adjacency is established with a neighbor

Syntax
```
ip ospf poll-interval time-seconds
no ip ospf poll-interval time-seconds?
```

Mode
Interface Configuration

Parameters

- **time-seconds** – the length of time in seconds, specified as an integer between 1 and 65535, inclusive
Description

The `ip ospf poll-interval` command specifies the length of time, in seconds, between OSPF packets that the router sends before adjacency is established with a neighbor. Utilizing this command reduces network overhead in cases where a router may have a neighbor on a given interface at the expense of initial convergence time.

The negative form of this command, `no ip ospf poll-interval`, removes the configured `time-seconds` value and returns this to its default value of 120 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip ospf poll-interval` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip ospf poll-interval 120
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following configures the poll-interval for interface fxp0 to be 100 seconds.

```
(config)# interface fxp0
(config-if)# ip ospf poll-interval 100
(config-if)# exit
(config)#
```

See Also

"poll-interval" ("poll-interval" on page 186)
"area poll-interval" ("area poll-interval" on page 210)

**ip ospf priority**

Name

`ip ospf priority` - specifies the priority for becoming the designated router (DR)

Syntax

```
ip ospf priority level
no ip ospf priority level?
```

Mode

Interface Configuration

Parameters

`level` - a priority number between 0 and 255, inclusive, for becoming a DR

Description

The `ip ospf priority` command specifies the priority on this interface for becoming the DR. This priority overrides any that is configured at the global or area levels. (See "priority" ("priority" on page 187) and "area priority" ("area priority" on page 211).) When more than one router attached to a network attempts to become the DR, the one with the highest priority wins. If the competing routers have the same priority, the one with the highest router ID becomes the DR. The router coming in second in the election becomes the backup DR. A router with a priority set to 0 is ineligible to become the DR.
The negative form of this command, `no ip ospf priority`, removes the configured priority. **Note:** Specifying a value for `level` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Note:** This command applies only to broadcast or NBMA media.

**Default**

If the `ip ospf priority` command is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if)# ip ospf priority 1
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures an area priority of 10. It also configures the priority for interface fxp1 to be 5.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 priority 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf priority 5
(config-if)# exit
(config)#
```

**See Also**

"priority" ("priority" on page 187)
"area priority" ("area priority" on page 211)

### ip ospf retransmit-interval

**Name**

`ip ospf retransmit-interval` - specifies the number of seconds between link state advertisement (LSA) retransmissions for adjacencies

**Syntax**

```
ip ospf retransmit-interval time-seconds
no ip ospf retransmit-interval time-seconds?
```

**Mode**

Interface Configuration

**Parameters**

`time-seconds` - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

**Description**

The `ip ospf retransmit-interval` command sets the default for the number of seconds between LSA retransmissions for adjacencies. If a Link State Protocol (LSP) is not acknowledged within the number of seconds specified here, it is re-sent.
This command overrides any configured in the global or area levels. (See "retransmit-interval" ("retransmit-interval" on page 195) and "area retransmit-interval" ("area retransmit-interval" on page 213).)

The negative form of this command, no ip ospf retransmit-interval, removes the configured retransmit value. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If the ip ospf retransmit-interval command is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if)# ip ospf retransmit-interval 5
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the retransmit interval for area 1.2.3.4 to be 10 seconds. This value is then overridden in interface fxp1, where it is configured to be 7 seconds.

```
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 retransmit-interval 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf retransmit-interval 7
(config-if)# exit
(config)#
```

**See Also**
"retransmit-interval" ("retransmit-interval" on page 195)
"area retransmit-interval" ("area retransmit-interval" on page 213)

---

**ip ospf traffic-eng administrative-weight**

**Name**
`ip ospf traffic-eng administrative-weight` - sets the cost of the interface for Traffic Engineering purposes

**Syntax**
```
ip ospf traffic-eng administrative-weight number
no ip ospf traffic-eng administrative-weight number?
```

**Mode**
Interface Configuration

**Parameters**
`number` - an integer from 1 to 65535, inclusive

**Description**
Use the `ip ospf traffic-eng administrative-weight` command to configure the cost of the interface for Traffic Engineering purposes. This cost can be different than the normal OSPF interface cost.
The negative form of this command, `no ip ospf traffic-eng administrative-weight`, removes the configured `number` value.

**Notes:**
- Specifying a value for `number` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).

**Default**
This command is not explicitly configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures a traffic engineering cost of 6 on interface fxp2.

```
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf traffic-eng administrative-weight 6
```

**See Also**
"enable-te" ("enable-te" on page 179)

```
ip ospf traffic-eng attribute-flags
```

**Name**
ip ospf traffic-eng attribute-flags – sets the OSPF Traffic Engineering attributes for the interface

**Syntax**
```
ip ospf traffic-eng attribute-flags number
no ip ospf traffic-eng attribute-flags number?
```

**Mode**
Interface Configuration

**Parameters**
`number` – an integer from 1 to 4,294,967,295, inclusive

**Description**
Use the `ip ospf traffic-eng attribute-flags` command to sets the OSPF Traffic Engineering attributes for the interface. These flags can be used to indicate which link groups (or "colors") are present on the interface. The negative form of this command, `no ip ospf traffic-eng attribute-flags`, removes the configured `number` value.

**Notes:**
- Specifying a value for `number` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).

Default
This command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a traffic engineering attribute flag of 1 on interface fxp2.

```
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf traffic-eng attribute-flags 1
```

See Also
"enable-te" ("enable-te" on page 179)

**ip ospf traffic-eng bandwidth**

**Name**
ip ospf traffic-eng bandwidth - sets the bandwidth of the interface

**Syntax**

```
ip ospf traffic-eng bandwidth kbps
no ip ospf traffic-eng bandwidth kbps
```

**Mode**
Interface Configuration

**Parameters**

`kbps` - an integer from 1 to 65535, inclusive, specifying a number of kilobytes per second

**Description**
The **ip ospf traffic-eng bandwidth** command configures the bandwidth of the interface for Traffic Engineering information distributed within the OSPF domain. The negative form of this command, **no ip ospf traffic-eng bandwidth**, removes the configured `kbps` value.

Notes:

- Specifying a value for `kbps` in the **no** form has no effect on the configuration. Thus, it is displayed above as optional.
- Traffic engineering must be enabled on the router before this command will take effect. See "enable-te" ("enable-te" on page 179).

**Default**
This command is not explicitly configured by default.
Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a traffic engineering bandwidth of 1000 kbps for interface fxp2.

```
(config)# router ospf 1
(config-router-ospf)# enable-te
(config-router-ospf)# exit
(config)# interface fxp2
(config-if)# ip ospf traffic-eng bandwidth 1000
```

See Also
"enable-te" ("enable-te" on page 179)

**ip ospf transmit-delay**

**Name**

`ip ospf transmit-delay` - specifies the number of seconds required to transmit a link state update

**Syntax**

```
ip ospf transmit-delay time-seconds
no ip ospf transmit-delay time-seconds?
```

**Mode**

Interface Configuration

**Parameters**

`time-seconds` - the length of time in seconds, between 1 and 65535, inclusive specified as an integer

**Description**

The `ip ospf transmit-delay` command sets the estimated number of seconds required to transmit a link state update. This command takes into account transmission and propagation delays and must be greater than 0. This command overrides any given at the global or area levels ("transmit-delay" ("transmit-delay" on page 200) and "area transmit-delay" ("area transmit-delay" on page 216)).

The negative form of this command, `no ip ospf transmit-delay`, removes the configured `time-seconds` value. **Note:** Specifying a value for `time-seconds` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip ospf transmit-delay` is not specified, then this value defaults to the configured area or global value, respectively. If neither of those is configured, then this value defaults to the following:

```
(config-if)# ip ospf transmit-delay 1
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the transmit-delay for area 1.2.3.4 to be 10 seconds. This value is then overridden on interface fxp1, where it is configured to be 5 seconds.
(config)# router ospf 1
(config-router-ospf)# area 1.2.3.4 transmit-delay 10
(config-router-ospf)# exit
(config)# interface fxp1
(config-if)# ip ospf transmit-delay 5
(config-if)# exit
(config)#

See Also
"transmit-delay" ("transmit-delay" on page 200)
"area transmit-delay" ("area transmit-delay" on page 216)

show ip ospf

Name
show ip ospf - displays information about a single OSPF instance or all OSPF instances

Syntax
show ip ospf instance_id?

Mode
User Execution

Parameters
instance_id - specifies to return information for a specific instance. This can be an integer from 1 to 4,294,967,294, inclusive. Entering this is optional.

Description
Use show ip ospf to obtain information about a specific OSPF instance or all OSPF instances.

This query has two forms. If the query is issued without an instance ID, then information on about all OSPF instances is returned. Alternatively, the query can be issued with a specific instance ID. If this is the case, then the reply will contain information pertaining only to that referenced instance.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example is a request for information relating to OSPF instance 2.

> show ip ospf 2
Routing Instance "2" with ID 192.168.10.1
RFC2328 Compliant
RFC1583 mode is enabled
Configured as an area border and autonomous system boundary router
Redistributing External Routes from:
Static
BGP
Number of areas in this router is 1
Area 1.1.1.1
Number of interfaces in this area is 1
Configured as an NSSA area
Area has simple password authentication
SPF algorithm is executed 3 times
default-cost is 5
nssa-translate-always is enabled
nssa-translator-stability-interval is disabled
area range is 192.10.0.0/24
default-metric is 10
default-nssa-type is 5

**show ip ospf border-routers**

**Name**
show ip ospf border-routers - displays the Autonomous System (AS) and Area Border Routers (ABRs) for all OSPF instances

**Syntax**
```
show ip ospf border-routers
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use show ip ospf border-routers to display the AS and ABRs for all OSPF instances.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example is a request border router information.
```
> show ip ospf border-routers
OSPF Routing Table
Route 192.168.10.1 area 0.0.0.1 ABR
```

**show ip ospf database**

**Name**
show ip ospf database - displays information contained within the OSPF Link State Database(s)

**Syntax**
```
show ip ospf database [instance_id]
show ip ospf database instance_id area_id
```

**Mode**
User Execution
Parameters

*instance_id* - specifies to return information for a specific instance. This can be an integer from 1 to 4,294,967,294, inclusive. Entering this is optional.

*area_id* - if an instance ID is specified for a database, you can also optionally specify an area to be associated with this query.

Description

The `show ip ospf database` query displays the information contained in the OSPF Link State Database(s). This query can be narrowed to search a specific database, based on the instance ID, the area ID, whether to search the external database or the type-11-opaque database.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example shows a request for all OSPF database information.

```
> show ip ospf database
OSPF Router Instance 1
Router Link States (Area 0.0.0.1)
Link ID        ADV RTR     Seq No  Age Checksum  Link ct
192.168.10.0  192.168.1.8 0x80000003  0    0xBEEF     1
192.168.11.0  192.168.2.4 0x80000003  0    0xBEED     2
```

**show ip ospf interface**

**Name**

`show ip ospf interface` - displays information about a single or all interfaces running OSPF

**Syntax**

```
show ip ospf interface [instance_id interface-name]?
```

**Mode**

User Execution

**Parameters**

*instance_id interface-name* - optionally specify an instance ID along with a physical interface name to view information for a specific interface

**Description**

Use `show ip ospf interface` to obtain information about a specific interface or all interfaces on which OSPF is running.

This query has two forms. If the query is issued without arguments, then information on about all interfaces over which OSPF is running is returned. Alternatively, the query can be issued with a specific interface. If this is the case, then the reply will contain information pertaining only to that referenced interface. **Note:** When querying for a specific interface, you must first include the instance ID.

Note that if interfaces are added to the set of interfaces over which OSPF is running after this query has been issued but before the query is finished, it is not guaranteed that the new interfaces will be reported. Similarly, if OSPF is de-configured on an interface after the query has been issued but before the query is finished, then the interface may or may not be reported. Finally, if a machine is in idle state, OSPF interface information will not display.
Command History
NGC 2.2 - This command was introduced.

Examples
The following example returns OSPF information for interface fxp0 in instance 1.

```
> show ip ospf interface 1 fxp0
fxp0 is up, line protocol is up
Internet Address 192.168.10.1/24, Area 0.0.0.1
Router ID 192.168.10.5, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, state WAITING, Priority 1
Designated Router is 192.168.10.5 (this router)
No Backup Designated router for this network
Timer intervals configured, Hello 20, Dead, 120, Wait 30,
Retransmit 5
Hello due in 0:00:05
Neighbor Count is 0, Adjacent neighbor count is 0
```

show ip ospf neighbor

Name
```
show ip ospf neighbor - displays information about OSPF neighbors
```

Syntax
```
show ip ospf neighbor neighbor?
```

Mode
User Execution

Parameters
```
neighbor - optionally specify either a valid IPv4 address in dotted-quad notation or a physical interface
name for the OSPF neighbor on which to query information
```

Description
The show ip ospf neighbor query displays information about a single OSPF neighbor or all OSPF neighbors.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example returns OSPF information for neighbor ID 2.2.2.2.

```
> show ip ospf neighbor 2.2.2.2
Routing Process "ospf 1":
Neighbor 0.0.103.12, interface address 192.0.0.2
In area 0.0.0.2 interface "fxp0"
Neighbor priority is 1, state is "Full", 7 state changes
DR is 192.0.0.2 BDR is 192.0.0.1
Options is 0x2
Dead timer is due in 4 seconds
```
**show ip ospf request-list**

**Name**
show ip ospf request-list - displays a list of all LSAs requested by a router

**Syntax**
```
show ip ospf request-list [ neighbor | interface | interface_neighbor ]?
```

**Mode**
User Execution

**Parameters**
- `neighbor` - optionally specify to display a list of all LSA requested by the router from this neighbor
- `interface` - optionally specify to display a list of all LSAs requested by the router from this interface
- `interface_neighbor` - optionally specify to display a list of all LSAs requested from this neighbor on this interface

**Description**
The `show ip ospf request-list` query displays a list of all LSAs requested by a router.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example displays a list of LSAs requested by the router for interface fxp0.

```
> show ip ospf request-list fxp0
OSPF Instance 1
Neighbor 192.168.10.5, interface fxp0 address 192.168.10.1

            Type    LS ID        ADV RTR      Seq No   Age  Checksum
1   192.168.10.9  192.168.10.9  0x80000001  35   0xBEEF
```

**show ip ospf retransmission-list**

**Name**
show ip ospf retransmission-list - displays a list of all LSAs waiting to be re-sent

**Syntax**
```
show ip ospf retransmission-list [ neighbor | interface | interface_neighbor ]?
```

**Mode**
User Execution

**Parameters**
- `neighbor` - optionally specify to display a list of all LSA requested by the router from this neighbor
- `interface` - optionally specify to display a list of all LSAs requested by the router from this interface
**interface_neighbor**— optionally specify to display a list of all LSAs requested from this neighbor on this interface

**Description**

The `show ip ospf retransmission-list query` displays a list of all LSAs requested by a router that are waiting to be re-sent.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example displays retransmission list information for interface fxp0.

```
> show ip ospf retransmission-list fxp0
OSPF Instance 1
Neighbor 192.168.10.5, interface fxp0 address 192.168.10.1
LSA retransmission due in 3000 msec, Queue length is 3

Type    LS ID       ADV RTR      Seq No    Age  Checksum
1   192.168.10.9  192.168.10.9  0x80000001  35   0xBEEF
```

---

**show ip ospf summary-address**

**Name**

`show ip ospf summary-address`— displays information about configured OSPF area ranges

**Syntax**

```
show ip ospf summary-address
```

**Mode**

User Execution

**Parameters**

none

**Description**

Use `show ip ospf summary-address` to obtain information about configured OSPF area ranges.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example queries area range information.

```
> show ip ospf summary-address
OSPF Instance 1
   192.168/20 Cost 0
```
show ip ospf virtual-links

Name
show ip ospf virtual-links - displays information about configured OSPF virtual links for a single instance or for all OSPF instances

Syntax

show ip ospf virtual-links

Mode
User Execution

Parameters
none

Description
Use show ip ospf virtual-links to obtain information about configured OSPF virtual links.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example queries virtual link information.

> show ip ospf virtual-links
Virtual Link to router 192.168.101.2 is up
Transit area 0.0.0.1 via interface fxp0
Link Cost is 5
Adjacency State is FULL
Timer intervals configured, Hello 20, Dead 120, Wait 30
Retransmit 5
Hello due in 0:00:05
Chapter 11

Redirect Processing

In This Chapter

Redirect Processing Overview 266
ip redirect 266
router redirect 267
trace file 267
trace flag 268

Redirect Processing Overview

ICMP redirects are messages sent by a router to an originator of data, indicating that a different hop should be used to reach the destination. A router sends a redirect when a routing table lookup for a received datagram results in transmission of the datagram out the same interface on which it was received.

ip redirect

Name

ip redirect - specifies that redirects received via the specified interfaces will be ignored

Syntax

ip redirect enable
no ip redirect

Mode

Interface Configuration

Parameters

none

Description

Use the ip redirect command to specify that redirects received via the specified interface should be ignored. The negative form of this command, no ip redirect, specifies that the redirects should be accepted.

Default

Redirects are accepted by default. Therefore, if ip redirect is not specified, it is the same as if the user had specified the following:

(config-if)# ip redirect enable

Command History

NGC 2.3 - This command was introduced.
Examples
The following example specifies to ignore redirects on interface fxp1.

```
(config)# interface fxp1
(config-if)# no ip redirect
(config-if)# exit
(config)#
```

**router redirect**

**Name**
router redirect - configures the Redirect protocol on the router

**Syntax**

```
router redirect
no router redirect
```

**Mode**
Global Configuration

**Parameters**
none

**Description**
Use the `router redirect` command to configure the Redirect protocol on the router. This also enters the user into Redirect Configuration mode, where tracing options can be specified. The negative form of this command, `no router redirect`, turns off the Redirect protocol.

**Default**
If `router redirect` is not specified, it is the same as if the user had specified the following:

```
(config)# no router redirect
```

**Command History**
NGC 2.3 - This command was introduced.

**Examples**
The following example specifies to ignore redirects on interface fxp1.

```
(config)# router redirect
(config-router-redirect)# exit
(config)#
```

**trace file**

**Name**
trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed
Syntax

trace file file_name [no-timestamp || overwrite]?

no trace file file_name [no-timestamp || overwrite]?

Mode
Redirect Router Configuration

Parameters

file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.

no-timestamp - specifies that a timestamp should not be prepended to all trace lines

overwrite - specifies to begin tracing by appending or truncating an existing file

Description

The trace file command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For Redirect, the trace file command in Redirect Router Configuration Mode specifies a file for tracing of all Redirect events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the trace flag command.

The no-timestamp option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The overwrite option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands.

Default

Redirect tracing options are not configured by default.

Command History

NGC 2.3 - This command was introduced.

Examples

In the following example, tracing is written to the file "rdirct.log". No timestamp will display at the beginning of the trace lines.

(config)# router redirect enable
(config-router-redirect)# trace file rdirct.log no-timestamp

trace flag

Name

trace flag - specifies redirect tracing options

Syntax

trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ other | packets ] [receive]? [detail?] )

no trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ other | packets ] [receive]? [detail?] )

Mode

Redirect Router Configuration
Parameters
Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ]
```
- These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
  - `route` - trace routing table changes for routes installed by this protocol or peer
  - `normal` - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
  - `state` - trace state machine transition in the protocol
  - `policy` - trace the application of protocol and user-specified policy to routes being imported or exported
  - `task` - trace system interface and processing associated with this protocol
  - `timer` - trace timer usage by this protocol
  - `all` - turns on all trace flags

Redirect-specific flags:

```
[ other | packets ]
```
- These Redirect-specific flags can be associated with the receive and detail action items. These flags are defined as follows:
  - `other` - trace all other Redirect messages.
  - `packets` - trace all Redirect packets

[receive]?
- optionally specify whether to limit the tracing to received packets

[detail]?
- optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

Description
Use the `trace flag` command to specify Redirect tracing flags. Each flag must reside on its own configuration line. For example, you cannot specify to trace both task and policy packets in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.3 - This command was introduced.

Examples
In the following example, flags specify to trace both timer usage and state machine transition. This tracing information will be written to the file rdirct.log.

```
(config)# router redirect enable
(config-router-redirect)# trace file rdirct.log
(config-router-redirect)# trace flag timer
(config-router-redirect)# trace flag state
```
Chapter 12

Router Discovery

In This Chapter

- Router Discovery Overview: 270
- ip router-discovery address-policy: 271
- ip router-discovery enable: 272
- ip router-discovery trace file: 272
- ip router-discovery trace flag: 273
- router-discovery lifetime: 274
- router-discovery maximum-interval: 275
- router-discovery minimum-interval: 276

Router Discovery Overview

The Router Discovery Protocol is an IETF standard protocol (RFC 1256) used to inform hosts of the existence of routers. It is intended to be used instead of having hosts wiretap routing protocols, such as RIP. It is used in place of, or in addition to, statically-configured default routes in hosts. RFC 1256 can be found at: http://ietf.org/rfc/rfc1256.txt. The protocol is split into two portions: the server portion, which runs on routers, and the client portion, which runs on hosts. Advanced Routing Suite currently supports only the server configuration.

The router discovery server runs on routers and announces their existence to hosts by periodically multicasting or broadcasting a router advertisement to each interface on which it is enabled. These router advertisements contain a list of all the routers' addresses on a given interface and the preference of each address for use as the default router on that interface.

Initially, these router advertisements occur every few seconds, then fall back to every few minutes. In addition, a host can send a router solicitation to which the router will respond with a unicast router advertisement (unless a multicast or broadcast advertisement is due momentarily).

Each router advertisement contains an advertisement "lifetime", indicating for how long the advertised addresses are valid. This lifetime is configured such that another router advertisement will be sent before the lifetime has expired. A lifetime of zero is used to indicate that one or more addresses are no longer valid.

On systems supporting IP multicasting, the router advertisements are, by default, sent to the all-hosts multicast address, 224.0.0.1. However, an advertisement method of broadcast can be specified. When router advertisements are being sent to the all-hosts multicast address, or an interface is configured for the limited-broadcast address, 255.255.255.255, all IP addresses configured on the physical interface are included in the router advertisement. When the router advertisements are being sent to a net or subnet broadcast, only the address associated with that net or subnet is included.

A host listens for router advertisements via the all-hosts multicast address (224.0.0.1) if IP multicasting is available and enabled, or on the interface's broadcast address. When starting up, or when reconfigured, a host can send a few router solicitations to the all-routers multicast address, 224.0.0.2, or the interface's broadcast address.

When a router advertisement with non-zero lifetime is received, the host installs a default route to each of the advertised addresses. If the preference is ineligible, or the address is not on an attached interface, the route is marked unusable but is retained. If the preference is usable, the metric is set as a function of the preference such that the route with the best preference is used. If more than one address with the same preference is received, the one with the lowest IP address will be used. These default routes are not exportable to other protocols.
When a router advertisement with a zero lifetime is received, the host deletes all routes with next-hop addresses learned from that router. In addition, any routes learned from ICMP redirects pointing to these addresses will be deleted. The same will happen when a router advertisement is not received to refresh these routes before the lifetime expires.

**ip router-discovery address-policy**

**Name**

`ip router-discovery address-policy` - specifies policy-related parameters when configuring logical interface addresses in router discovery.

**Syntax**

```
ip router-discovery address-policy ipv4_address
[ (method (broadcast | ignore | multicast)) |
(preference pref) | ineligible]
no ip router-discovery address-policy ipv4_address
```

**Mode**

Global Configuration

**Parameters**

- `ipv4_address` - a valid IPv4 address in dotted-quad notation
- `method (multicast | broadcast | ignore)` - optionally specify whether the associated IPv4 address should be included in router advertisements and whether the router advertisements should be unicast or multicast. If ignore is specified, then the associated IPv4 address will not be included in router advertisements. Note: If an interface does not support multicast, but multicast is specified for that interface, then that interface will not be advertised.
- `preference pref` - configures the preferability of the address as a default router address. Otherwise, you can specify a value for `pref` between 1 and 255. The default preference value is 55.
- `ineligible` - optionally specify that the router should be advertised with the worst possible preference, making it ineligible to become the default router

**Description**

Use the `ip router-discovery address-policy` command to specify the logical interface or interfaces that should be advertised in router advertisements. This command can be given multiple times, creating multiple interfaces to be advertised in router advertisements.

**Default**

The `ip router-discovery address-policy` command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures the logical interface 223.1.5.8 with an advertisement method of broadcast and a preference of 10.

```
(config)# ip router-discovery address-policy 223.1.5.8
method broadcast preference 10
(config)#
```
**Example 2**
The following example configures the logical interface 224.5.5.5 with an advertisement method of multicast. This interface is then configured such that it is ineligible to become the default router.

```
(config)# ip router-discovery address-policy 224.5.5.5
method multicast ineligible
(config)#
```

**ip router-discovery enable**

**Name**
ip router-discovery enable - enables the Router Discovery server

**Syntax**
```
ip router-discovery enable
no ip router-discovery enable
```

**Mode**
Global Configuration

**Parameters**
none

**Description**
Use the `ip router-discovery enable` command to enable Router Discovery on the server. The negative form of this command disables Router Discover on the server.

**Default**
If `ip router-discovery enable` is not specified, it is the same as if the user had specified the following:
```
(config)# ip router-discovery enable
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example turns Router Discovery off on the server
```
(config)# no ip router-discovery enable
(config)#
```

**ip router-discovery trace file**

**Name**
ip router-discovery trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed

**Syntax**
```
ip router-discovery trace file file_name [no-timestamp || overwrite]?
no ip router-discovery trace file file_name [ no-timestamp ||overwrite]?
```
Mode
Global Configuration

Parameters

file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.

no timestamp - specifies that a timestamp should not be prepended to all trace lines

overwrite - specifies to begin tracing by appending or truncating an existing file

Description
The trace file command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For Router Discovery, the ip router-discovery trace file command specifies a file for tracing of all Router Discovery events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the ip router-discovery trace flag command.

The no timestamp option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The overwrite option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands.

Default
Router Discovery tracing is turned off by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, Router Discovery tracing is written to the file "rdisc.log". No timestamp will display at the beginning of the trace lines.

   (config)# ip router-discovery trace file rdisc.log no-timestamp

ip router-discovery trace flag

Name
ip router-discovery trace flag - specifies tracing options that are common across all protocols

Syntax

   ip router-discovery trace flag ( [ route | normal | state | policy | task | timer | all ] )

   no ip router-discovery trace flag ( [ route | normal | state | policy | task | timer | all ] )

Mode
Global Configuration

Parameters

Flags common to all protocols:

[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item.
Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. Note: Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

### Description

Use the `ip router-discovery trace flag` command to specify tracing flags for Router Discovery tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both task and policy packets in the same command.

### Default

The default is for no flags to be explicitly configured.

### Command History

NGC 2.2 - This command was introduced.

### Examples

In the following example, flags specify to trace state machine transitions. This tracing information will be written to the file `rdisc.log`.

```
(config)# ip router-discovery trace file rdisc.log
(config)# ip router-discovery trace flag state
```

---

**router-discovery lifetime**

### Name

`router-discovery lifetime` - specifies the length of time that the addresses in a given router advertisement are valid

### Syntax

```
router-discovery lifetime time-seconds
no router-discovery lifetime time-seconds?
```

### Mode

Interface Configuration

### Parameters

- `time-seconds` - specifies the length of time, in seconds, with a value between the Maximum Advertisement Interval value (which has a minimum value of 4) and 9,000, inclusive

### Description

The `router-discovery lifetime` command specifies how long the addresses in a given router advertisement are valid. Lifetime must be no less than "router-discovery maximum-interval" ("router-discovery maximum-interval" on page 275) and no more than 2 hours and 30 minutes. If this value is set to
less than the Maximum Advertisement Interval, then Advanced Routing Suite will reconfigure `router-discovery lifetime` to be equal to the Maximum Advertisement Interval.

The negative form of this command, `no router-discovery lifetime`, removes the configured time-seconds value and returns this to its default value. **Note:** Specifying a value for time-seconds in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `router-discovery lifetime` is not specified, it will default to 3 times the value of the Maximum Advertisement Interval. Therefore, if neither value is configured, then the `router-discovery lifetime` will default to:

```
(config-if)# router-discovery lifetime 1800
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example runs the router discovery server on interface fxp0. All routers that it advertises out of interface fxp0 will be advertised with a lifetime of 10 minutes.

```
(config)# interface fxp0
(config-if)# router-discovery lifetime 1800
(config-if)# exit
(config)#
```

**See Also**

"router-discovery maximum-interval" ("`router-discovery maximum-interval`" on page 275)
"router-discovery minimum-interval" ("`router-discovery minimum-interval`" on page 276)

---

**router-discovery maximum-interval**

**Name**

`router-discovery maximum-interval` - specifies the maximum time allowed between sending unsolicited router advertisements from the interface

**Syntax**

```
router-discovery maximum-interval time-seconds
no router-discovery maximum-interval time-seconds?
```

**Mode**

Interface Configuration

**Parameters**

`time-seconds` - specifies the length of time, in seconds, with a value between 4 and 1800, inclusive

**Description**

The `router-discovery maximum-interval` command specifies the maximum time allowed between sending unsolicited router advertisements from the interface. This must be no less than 4 seconds and no more than 30 minutes (1800 seconds).

The negative form of this command, `no router-discovery maximum-interval`, removes the configured time-seconds value and returns this to its default value of 600 seconds. **Note:** Specifying a value for time-seconds in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
Default
If \texttt{router-discovery maximum-interval} is not specified, it is the same as if the user had specified the following:
\begin{verbatim}
(config-if)# router-discovery maximum-interval 600
\end{verbatim}

Command History
NGC 2.2 - This command was introduced.

Examples
The following example runs the router discovery server on interface \texttt{fxp0}, sending advertisements no more often than once every 6 minutes.
\begin{verbatim}
(config)# interface fxp0
(config-if)# router-discovery maximum-interval 360
(config-if)# exit
(config)#
\end{verbatim}

See Also
"router-discovery lifetime" ("router-discovery lifetime" on page 274)
"router-discovery minimum-interval" ("router-discovery minimum-interval" on page 276)

\textbf{router-discovery minimum-interval}

Name
\texttt{router-discovery minimum-interval} - specifies the minimum time allowed between sending unsolicited router advertisements from the interface

Syntax
\begin{verbatim}
router-discovery minimum-interval time-seconds
no router-discovery minimum-interval time-seconds?
\end{verbatim}

Mode
Interface Configuration

Parameters
\begin{itemize}
  \item \texttt{time-seconds} - specifies the length of time, in seconds, with a value between 3 and "router-discovery maximum-interval" ("router-discovery maximum-interval" on page 275) (which has a maximum value of 1800), inclusive
\end{itemize}

Description
The \texttt{router-discovery minimum-interval} command specifies the minimum time allowed between sending unsolicited router advertisements from the interface. This must be no less than 3 seconds and no more than the "router-discovery maximum-interval" ("router-discovery maximum-interval" on page 275) value. If this value is set to greater than the Maximum Advertisement Interval, Advanced Routing Suite will reconfigure the value of the Minimum Advertisement Interval to be equal to the value of the Maximum Advertisement Interval log the change.

The negative form of this command, \texttt{no router-discovery minimum-interval}, removes the configured time-seconds value and returns this to its default value. Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.
Default

If `router-discovery minimum-interval` is not specified, it will default to 3/4 the value of the Maximum Advertisement Interval. Therefore, if neither value is configured, then the `router-discovery minimum-interval` will default to:

```
(config-if)# router-discovery minimum-interval 450
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example runs the router discovery server on interface fxp0, sending advertisements no less often than once every minute.

```
(config)# interface fxp0
(config-if)# router-discovery minimum-interval 60
(config-if)# exit
(config)#
```

See Also

"router-discovery lifetime" ("router-discovery lifetime" on page 274)
Chapter 13

Routing Information Protocol (RIP)

In This Chapter

Routing Information Protocol (RIP) Overview 278
router rip 281
default-metric 281
distribute-list 282
ecmp 285
enable 285
flash-update-time 286
ignore-host-routes 287
ignore-must-be-zero 287
network 288
preference 289
query-authentication 290
redistribute 291
send-updates 293
source-gateways 294
split-horizon 296
term-updates 297
timers basic 298
trace file 299
trace flag 300
trusted-gateways 301
ip rip authentication 302
ip rip enable 304
ip rip metric-in 304
ip rip metric-out 305
ip rip no-receive 306
ip rip no-send 307
ip rip secondary-authentication 308
ip rip version 309
show ip rip database 310

Routing Information Protocol (RIP) Overview

One of the most widely used interior gateway protocols is the Routing Information Protocol (RIP). RIP is an implementation of a distance-vector, or Bellman-Ford, algorithm. RIP classifies routers as active and passive (silent). Active routers advertise their routes (reachability information) to others; passive routers listen and update their routes based on advertisements, but do not advertise. Typically, routers run RIP in active mode, while hosts use passive mode.

A router running RIP in active mode sends updates at set intervals. Each update contains paired values, where each pair consists of an IP network address and an integer distance to that network. RIP uses a hop count metric to measure the distance to a destination. In the RIP metric, a router advertises directly
Routing Information Protocol (RIP)

connected networks at a metric of 1 by default. Networks that are reachable through one other gateway are 2 hops, etc. Thus, the number of hops or hop count along a path from a given source to a given destination refers to the number of gateways that a datagram would encounter along that path. Using hop counts to calculate shortest paths does not always produce optimal results. For example, a path with a hop count 3 that crosses three Ethernets may be substantially faster than a path with a hop count 2 that crosses two slow-speed serial lines. To compensate for differences in technology, many routers advertise artificially high hop counts for slow links.

RIP dynamically builds on information received through RIP updates. When started up, RIP issues a request for routing information and then listens for responses to the request. If a system configured to supply RIP hears the request, it responds with a response packet based on information in its routing database. The response packet contains destination network addresses and the routing metric for each destination.

When a RIP response packet is received, the routing daemon takes the information and rebuilds the routing database, adding new routes and "better" (lower metric) routes to destinations already listed in the database. RIP also deletes routes from the database if the next router to that destination reports that the route contains more than 15 hops, or if the route is deleted. All routes through a gateway are deleted if no updates are received from that gateway for a specified time period. In general, routing updates are issued every 30 seconds. In many implementations, if a gateway is not heard from for 180 seconds, all routes from that gateway are deleted from the routing database. This 180-second interval also applies to deletion of specific routes.

RIP version 2 (more commonly known as RIP II) adds additional capabilities to RIP. Some of these capabilities are compatible with RIP I and some are not. To avoid supplying information to RIP I routes that could be misinterpreted, RIP II can use only non-compatible features when its packets are multicast. On interfaces that are not capable of IP multicast, RIP-I-compatible packets are used that do not contain potentially confusing information.

Some of the most notable RIP II enhancements are:

- Next hop
- Network mask
- Authentication
- RIP tag field

These features in RIP I and II are contrasted in the following paragraphs.

Next hop

With RIP II, a router can advertise a next hop other than itself. Next hop is useful when advertising a static route to a dumb router that does not run RIP, because it avoids having packets that are passed through the dumb router from having to cross a network twice. Because RIP I routers will ignore next hop information in RIP II packets, packets might cross a network twice, which is exactly what happens with RIP I. Next hop information is provided in RIP-I-compatible RIP II packets.

Network mask

RIP I assumes that all subnetworks of a given network are classful (Class A,B,C). RIP I uses this assumption to calculate the network masks for all routes received. This assumption prevents subnets with classless netmasks from being included in RIP packets. RIP II adds the ability to specify the network mask with each network in a packet. Because RIP I routers will ignore the network mask in RIP II packets, their calculation of the network mask will quite possibly be wrong. For this reason, RIP-I-compatible RIP II packets must not contain networks that would be misinterpreted. These networks must be provided only in native RIP II packets that are multicast.

RIP I derives the network mask of received networks and hosts from the network mask of the interface via which the packet was received. If a received network or host is on the same natural network as the interface over which it was received, and that network is subnetted (the specified mask is more or less specific than the natural netmask), the interface’s subnet mask is applied to the destination. If bits outside the mask are set, it is assumed to be a host; otherwise, it is assumed to be a subnet. On point-to-point interfaces, the netmask is applied to the remote address. The netmask on these interfaces is ignored if it matches the natural network of the remote address, or is all ones. Unlike previous releases, the zero subnet (a subnetwork that matches the natural network of the interface, but has a more specific, or longer, network mask) is advertised. If this is not desirable, a route filter may be used to reject it.
Authentication

RIP II packets may contain one of two types of authentication strings that may be used to verify the validity of the supplied routing data. Authentication may be used in RIP-I-compatible RIP II packets, but be aware that RIP I routers will ignore these packets (unless ignore-must-be-zero is configured off). The first method is a simple password in which an authentication key of up to 16 characters is included in the packet. If this key does not match what is expected, the packet will be discarded. This method provides very little security because it is possible to learn the authentication key by watching RIP packets.

The second method uses the MD5 algorithm to create a crypto-checksum of a RIP packet and an authentication key of up to 16 characters. The transmitted packet does not contain the authentication key itself; instead, it contains a crypto-checksum, called the "digest". The receiving router will perform a calculation using the correct authentication key and discard the packet if the digest does not match. In addition, a sequence number is maintained to prevent the replay of older packets. This method provides a much stronger assurance that routing data originated from a router with a valid authentication key.

Two authentication methods can be specified per interface. Packets are always sent using the primary method, but received packets are checked with both the primary and secondary methods before being discarded. In addition, a separate authentication key is used for non-router queries.

RIP Commands

Global Configuration Mode RIP Commands
"router rip" ("router rip" on page 281)

Router Mode RIP Commands
"default-metric" ("default-metric" on page 281)
"distribute-list" ("distribute-list" on page 282)
"ecmp" ("ecmp" on page 285)
"enable" ("enable" on page 285)
"flash-update-time" ("flash-update-time" on page 286)
"ignore-host-routes" ("ignore-host-routes" on page 287)
"ignore-must-be-zero" ("ignore-must-be-zero" on page 287)
"network" ("network" on page 288)
"preference" ("preference" on page 289)
"query-authentication" ("query-authentication" on page 290)
"redistribute" ("redistribute" on page 291)
"send-updates" ("send-updates" on page 293)
"source-gateways" ("source-gateways" on page 294)
"split-horizon" ("split-horizon" on page 296)
"term-updates" ("term-updates" on page 297)
"timers basic" ("timers basic" on page 298)
"trace file" ("trace file" on page 299)
"trace flag" ("trace flag" on page 300)
"trusted-gateways" ("trusted-gateways" on page 301)

RIP Interface Commands
"ip rip authentication" ("ip rip authentication" on page 302)
"ip rip enable" ("ip rip enable" on page 304)
"ip rip metric-in" ("ip rip metric-in" on page 304)
"ip rip metric-out" ("ip rip metric-out" on page 305)
"ip rip no-receive" ("ip rip no-receive" on page 306)
"ip rip no-send" ("ip rip no-send" on page 307)
"ip rip secondary-authentication" ("ip rip secondary-authentication" on page 308)
"ip rip version" ("ip rip version" on page 309)

**RIP Querying Commands**
"show ip rip database" ("show ip rip database" on page 310)
"show ip rip gateway-summary"

**router rip**

**Name**

router rip - enters the user into Router Configuration mode for RIP

**Syntax**

router rip

no router rip

**Mode**

Global Configuration

**Parameters**

none

**Description**

Use the router rip command to enter RIP Router Configuration mode. Once you are in RIP Router Configuration mode, you can begin configuring RIP.

**Default**

None. You must specify this command in order to configure RIP.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example shows how to enter RIP Router Configuration mode.

```
(config)# router rip
(config-router-rip)#
```

**default-metric**

**Name**

default-metric - configures the default metric for advertising RIP routes learned from other protocols

**Syntax**

default-metric metric_value

no default-metric metric_value?
Mode
RIP Router Configuration

Parameters

metric_value - an integer from 0 to 16, inclusive, assigned to exported reachability

Description
Use the default-metric command to set the default metric value for advertising RIP routes learned from other protocols. All routes exported into RIP will receive this metric unless a matching route-map sets the metric to a different value.

The negative form of this command, no default-metric, removes the configured value and returns this to its default value of 1. Note: Specifying a value for metric_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If default-metric is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# default-metric 1
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the RIP default metric value to be 2.

```
(config)# router rip
(config-router-rip)# default-metric 2
(config-router-rip)# exit
(config)#
```

distribute-list

Name
distribute-list - configures policy for RIP to apply to incoming or outgoing updates

Syntax

```
distribute-list access_list_name [in [interface-name || route-map rm_name]?] | [out [interface-name || protocol]? ]

no distribute-list access_list_name [in [interface-name || route-map rm_name]?] | [out [interface-name || protocol]? ]
```

Mode
RIP Router Configuration Mode

Parameters
distribute-list in

access_list_name - the name of an access list
in - specifies that the policy applies to incoming updates

interface-name - optionally specify a physical interface name to which this policy will apply. Otherwise, it applies to all interfaces.
route-map name - the name of a route map to apply to incoming routes. Specifying this is optional.

distribute-list out

access_list_name - the name of an access list

out - specifies that the policy applies to exported updates

interface-name - optionally specify a physical interface name to which this policy will apply. Otherwise, it applies to all interfaces.

protocol - optionally specify a redistributed protocol to filter. Valid values include: aggregate, bgp, isis, kernel, ospf, ospf-ase, rip, and static.

Description

The distribute-list command provides a policy filtering mechanism for RIP routes. If the distribute list configured in this command is specified with the in keyword, then the filter will apply to all imported routes (that is to say, routes learned from RIP neighbors). If the distribute list configured in this command is specified with the out keyword, then the filter applies to exported routes (that is to say, routes announced to RIP neighbors).

To delete a configured distribute list, use the negative form of the command. Note: All arguments of the original command must be supplied in order for the entry to be deleted.

distribute-list in

The distribute-list in command configures a RIP import policy (in other words, a policy for RIP to apply to incoming updates). You can specify both an interface-name and a Route Map in the policy. Note that order is not important when specifying both.

If an inbound distribute-list is configured, then each route received from other RIP neighbors will be evaluated against this inbound list. If the route matches the criteria specified by the inbound list, then the route is imported. If the route does not match the inbound list criteria, then the route is rejected. A route matches the inbound list criteria if it is permitted when evaluated against the referenced access list. Furthermore, if the inbound distribute-list references a route map, then the route must also match when evaluated against the route map.

If the distribute-list in command references a route map that has not yet been defined, then an empty route map is created with the specified name.

If two inbound distribute lists are configured, one that does not reference an interface and one that does, then the most specific inbound distribute list is used when evaluating a route received from other RIP neighbors. If the route was received over an interface not named in any inbound distribute list, then the inbound distribute list that does not reference an interface is used. Otherwise, the inbound distribute list that references the interface over which the route was received is used.

At most, one distribute-list in command can be specified without an interface. Subsequent distribute-list in commands that do not reference an interface will overwrite any such previous command. Similarly, at most one distribute-list in command can be specified for each interface, and subsequent commands that reference an interface will overwrite any such previous command.

distribute-list out

The distribute-list out command is used to configure export policy for RIP routes. Outbound distribute lists look similar to inbound distribute lists, except that they allow you to optionally specify an interface and a protocol, or more correctly, a route source. This allows you to filter exported routes based on whether they were learned from BGP, OSPF, OSPF-ASE or were aggregate, static, or kernel routes. If an interface is specified, then the export policy described by the distribute-list out command only applies to the referenced interface. At most one distribute-list out command can be specified for each interface for each protocol. Furthermore, at most one distribute-list out command that does not reference a protocol can be specified. If an outbound distribute list is configured without referencing a protocol, then when a route is being considered for export to RIP neighbors, it must be permitted by the access list referenced in the distribute list. If instead an outbound distribute list is configured that references a protocol, and if the route being considered for export originated from the referenced protocol, then the route must be permitted by the access list referenced in the distribute list. If more of both types of lists are present, then the route need only be permitted by the access list referenced by one of the outbound distribute lists.
It should be noted that an outbound distribute list has no effect in the absence of a "redistribute" ("redistribute" on page 291) command. Outbound distribute lists can be thought of as a way to further refine the export policy expressed with a "redistribute" ("redistribute" on page 291) command.

The negative form of this command, no distribute-list, removes the configured list. You must specify either in or out in the negative form. For outbound distribute lists, specifying a protocol is optional. If the protocol is not specified, then all distribute lists referencing the specified access list will be deleted.

**Default**
The default is for RIP to import everything.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
In the following example, incoming updates received on interface fxp0 are matched against a standard IP access-list named "abc".

```
(config)# router rip
(config-router-rip)# distribute-list abc in fxp0
(config-router-rip)# exit
(config)#
```

**Example 2**
In the following example, incoming updates received on interface fxp0 are matched against a standard IP access-list named "alist1". A route map named "rm1" is also applied to the incoming updates.

```
(config)# router rip
(config-router-rip)# distribute-list alist1 in route-map rm1 fxp0
(config-router-rip)# exit
(config)#
```

**Example 3**
In the following example, incoming updates from all interfaces are matched against a standard IP access-list named "alist1". A route map named "rm1" is also applied to incoming updates.

```
(config)# router rip
(config-router-rip)# distribute-list alist1 in route-map rm1
(config-router-rip)# exit
(config)#
```

**Example 4**
In the following example, the distribute-list entry originally configured in Example 3 is deconfigured.

```
(config)# router rip
(config-router-rip)# no distribute-list alist1 in route-map rm1
(config-router-rip)# exit
(config)#
```
**ecmp**

**Name**
ecmp - configures equal-cost multipaths

**Syntax**
- `ecmp`
- `no ecmp`

**Mode**
RIP Router Configuration

**Parameters**
none

**Description**
The `ecmp` command is used to enable the Equal-Cost Multi-Path (ECMP) feature. Use the negative form of this command, `no ecmp`, to disable the ECMP feature.

**Default**
If `ecmp` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# no ecmp
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example turns on the ECMP feature.
```
(config)# router rip
(config-router-rip)# ecmp
(config-router-rip)# exit
(config)#
```

**enable**

**Name**
enable - enables RIP on a router

**Syntax**
- `enable`
- `no enable`

**Mode**
RIP Router Configuration

**Parameters**
none
Description
The `enable` command enables the state RIP. The negative form of this command, `no enable`, disables RIP on the router.

Default
If `enable` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# enable
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example disables RIP on the router.

```
(config)# router rip
(config-router-rip)# no enable
(config-router-rip)# exit
(config)#
```

---

flash-update-time

Name
`flash-update-time` - sets the number of seconds between flash updates

Syntax
```
flash-update-time time-seconds
no flash-update-time time-seconds?
```

Mode
RIP Router Configuration

Parameters
`time-seconds` - an integer between 0 and 5, specifying a number of seconds

Description
The `flash-update-time` command configures the Flash Update time. By default, the flash-update-timer is set to a random value with a range of (0, 5) inclusive. The negative form of this command, `no flash-update-time`, removes the configured `time-seconds` value and returns this to a random number between 0 and 5. Note: Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default
The default Flash Update time value is a random number between 0 and 5.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the flash-update-timer to be 3 seconds.

```
(config)# router rip
```

---
Routing Information Protocol (RIP)

(config-router-rip)# flash-update-time 3
(config-router-rip)# exit
(config)#

**ignore-host-routes**

**Name**

`ignore-host-routes` - configures Advanced Routing Suite to reject host routes learned in RIP response messages

**Syntax**

```
ignore-host-routes
no ignore-host-routes
```

**Mode**

RIP Router Configuration

**Parameters**

none

**Description**

The `ignore-host-routes` command causes Advanced Routing Suite to reject host routes (with a netmask of 255.255.255.255) learned in RIP response messages. The negative form of this command, `no ignore-host-routes`, returns to the default setting of accepting host routes learned in RIP response messages.

**Default**

If `ignore-host-routes` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# no ignore-host-routes
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the router to not recognize host routes.

```
(config)# router rip
(config-router-rip)# ignore-host-routes
(config-router-rip)# exit
(config)#
```

**ignore-must-be-zero**

**Name**

`ignore-must-be-zero` - configures the handling of zero-filled reserved fields in RIP 1

**Syntax**

```
ignore-must-be-zero
```
no ignore-must-be-zero

**Mode**
RIP Router Configuration

**Parameters**
none

**Description**
The RIP version 1 specification mandates that certain fields in RIP routing updates are reserved and should be filled with zeros. It also mandates that version 1 packets, where the reserved fields are not filled with all zeros, should be discarded. However, some implementations of RIP do not follow the specification and carry non-zero information in these fields. The `ignore-must-be-zero` command allows RIP to interoperate with those implementations.

The user should be certain that the non-compliant router is intentionally generating malformed packets (rather than malfunctioning) before enabling interoperability. Otherwise, correct routing could be compromised.

The negative form of this command, `no ignore-must-be-zero`, returns this command to the default of checking zero-filled reserved fields for RIP version 1.

**Default**
If `ignore-must-be-zero` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# no ignore-must-be-zero
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the RIP router to ignore messages whose reserved fields are not zeroed.

```
(config)# router rip
(config-router-rip)# ignore-must-be-zero
(config-router-rip)# exit
(config)#
```

---

**network**

**Name**
`network` - configures a network on which to run RIP

**Syntax**
```
network ipv4_address [ mask ]? {1,n}
no network ipv4_address [ mask ]?
```

**Mode**
RIP Router Configuration

**Parameters**
- `ipv4_address` - a valid IPv4 address specified in dotted-quad notation
- `mask` - specify the mask (wildcard bits) of the network IP address. Specifying this value is optional.
The network command can be specified multiple times in RIP Router Configuration mode.

**Description**

The network command specifies a network on which to run RIP. Use the command multiple times to build a list of networks on which RIP is to run. If the optional mask is not specified, RIP will be enabled on all interfaces that match the natural mask of the network IP address, and that natural mask will display when you issue a show running command. If the mask is supplied, then RIP will run on interfaces that match both the network address and the mask.

Use the negative form of this command, no network, to stop running RIP on a network. Note: The complete configuration must be provided in the negative command. For example, if you configured a network address and specified a mask, both values must appear in the negative command.

**Default**

RIP networks are not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures RIP to run on interface 192.168.121.1 and the interface associated with network 192.168.11.0.

```
(config)# router rip
(config-router-rip)# network 192.168.121.1
(config-router-rip)# exit
(config)#
```

---

**preference**

**Name**

preference - specifies how active routes that are learned from RIP will be selected, compared to routes learned from other protocols

**Syntax**

preference pref

no preference pref?

**Mode**

RIP Router Configuration

**Parameters**

pref - an integer from 0 (directly connected) to 255, inclusive

**Description**

Routers can learn multiple routes to the same destination. Each routing protocol handles this internally, so, for example, if RIP learns two different routes to the same destination, it will select the route with the lower metric. However, a router can learn two routes to the same destination from different routing protocols, and it must find a way to choose between them. Advanced Routing Suite uses a preference number to make this decision, choosing the route with the lower preference. Normally, RIP routes are preferred to routes from external routing protocols (for example, BGP or OSPF ASE) and to those that are generated or aggregated. RIP routes are less preferable than those learned from other intra-domain routing protocols and those that are statically configured. The preference command is used to change this ordering.
The negative form of this command, `no preference`, removes the configured `pref` value and returns this to its default value of 100. **Note:** Specifying a value for `pref` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `preference` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# preference 100
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a preference of 150 for RIP routes.

```
(config)# router rip
(config-router-rip)# preference 150
(config-router-rip)# exit
(config)#
```

**query-authentication**

**Name**

`query-authentication` - configures the authentication used by the ripquery utility

**Syntax**

```
query-authentication [ [simple key] | [md5 id_number md5_key (start-generate date_time || stop-generate date_time || start-accept date_time || stop-accept date_time)?] (0,4) ]
```

**Mode**

RIP Router Configuration

**Parameters**

- `simple key` - specifies simple (clear password) authentication. The value for `key` is specified by one to eight decimal digits (with a value between 0 and 255) separated by periods, a one- to eight-byte hexadecimal string preceded by 0x, or a one- to eight-character string.

- `md5 id_number md5_key [(start-generate date_time || (stop-generate date_time) || (start-accept date_time || (stop-accept date_time))] (0,4)` - specifies the authentication used for specifying md5 cryptographic authentication. The value for `id_number` is an integer with a value between 1 and 255, inclusive. The value for `md5_key` is a one- to sixteen-character string. The `start` and `stop` values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.

- `{0,4}` - up to 4 MD5 key values can be configured, provided each has a unique `md5_key` value.

**Description**

The ripquery utility was created for debugging RIP routers. This tool sends a RIP POLL packet, which is an extension undocumented in the RFC. The `query-authentication` command is used to check the incoming POLL packets. If the authentication matches, then RIP will reply with its full routing table. (It will not run split-horizon or poison reverse before replying.) If the authentication does not match, then the request will be discarded.
Although `query-authentication` uses the standard key format for passwords, the generate portions of the key are irrelevant because the key is used only to check incoming requests. Outing packets use whatever authentication is set up on the interface over which the POLL packet was received.

Simple and md5 keys are mutually exclusive. Multiple md5 key values can be configured for an md5 authentication provided each key has a unique `md5_key` value.

**Default**

Query authentication is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures a simple key password for Query Authentication.

```
(config)# router rip
(config-router-rip)# query-authentication simple abc
(config-router-rip)# exit
(config)#
```

**Example 2**

The following example configures md5 authentication with two key chains.

```
(config)# router rip
(config-router-rip)# query-authentication md5 1 abc
start-accept 2003-12-01.10.20 stop-accept 2003-12-11.10.20
start-generate 2003-12-01.10.20 stop-generate 2003-12-11.10.20
(config-router-rip)# query-authentication md5 2 abc
start-accept 2003-12-21.10.20 stop-accept 2003-12-11.10.20
start-generate 2003-12-21.10.20 stop-generate 2003-12-11.10.20
(config-router-rip)# exit
(config)#
```

**redistribute**

**Name**

`redistribute` - specifies routes to export to the current protocol

**Syntax**

```
redistribute protocol [route-map name]? (0,9)
no redistribute protocol [route-map name]?
```

**Mode**

RIP Router Configuration

**Parameters**

`protocol` - the protocol name whose routes you want to redistribute to the current protocol being configured. Valid protocols are aggregate, bgp, direct, isis, kernel, ospf, ospf-ase, rip, and static.
route-map name - the name of a route map to apply to these routes. Specifying this is optional.

{0,9} - although this command can be given multiple times, it can only be given once for each of the nine configurable protocols. In other words, if a redistribute command is given for a protocol and route map, and then given again for the same protocol with a different route map, the second configuration overrides the first.

**Description**

Use the `redistribute` command to specify routes to export into the current protocol (BGP, OSPF-ASE, RIP, or VRE). This command causes routes from the specified protocol to be considered for redistribution into the current protocol. Additionally, if a route map is specified, then routes from the specified protocol that match the named route map will be considered for redistribution into the current protocol. If the referenced route map has not yet been configured, then an empty route map is created with the specified name.

**Note:** Configuring this away from its default removes the implicitly configured default. You will have to go back and specify to redistribute RIP and/or direct routes after the first `redistribute` configuration in order to export those routes.

**Default**

The default is to export direct routes and RIP routes. Note that this is an implicit default that is wiped away with the first redistribute configuration.

**Example**

**Example 1**

In the following example RIP is configured to redistribute all BGP routes.

```
(config)# router rip
(config-router-rip)# redistribute bgp
(config-router-rip)# exit
(config)#
```

**Example 2**

The following example configures a community set, "set1", that permits AS:num 101:102. It then configures an extended community set "ext-set1", that permits Route Target AS:num 201:202.

```
(config)# ip community-set set1 permit 101:102
(config)# ip extcommunity-set ext-set1 permit rt 201:202
```

The two are then added to a community list, called "commlist1".

```
(config)# ip community-list commlist1 permit set1
(config)# ip community-list commlist1 permit ext-set1
```

The community list is then applied to a route map called "match-commlist1". If the route map matches BGP Community list "commlist1", then the metric for routes will be set to 20.

```
(config)# route-map match-commlist1
(config-route-map)# match community commlist1
(config-route-map)# set metric 20
(config-route-map)# exit
(config)#
```

Finally, the route map ("match-commlist1") is applied to BGP routes and exported into RIP.

```
(config)# router rip
(config-router-rip)# redistribute bgp route-map match-commlist1
(config-router-rip)# exit
(config)#
```
Example 3

In the following example, route map "abc" is configured with the following match criteria:

If a route matches interface "fxp1" and a pre-configured BGP Community labeled "bgpcomm1", then communities specified in community "com-set-1" will be added to the route, communities specified in community labeled "com-set-2" will be deleted from the route, and the metric of the route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match community-set bgpcomm1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
```

This route map is then applied to static routes and exported into RIP.

```
(config)# router rip
(config-router-rip)# redistribute static route-map abc
(config-router-rip)# exit
```

See Also

"distribute-list" ("distribute-list" on page 282)

send-updates

Name

send-updates - specifies the conditions under which RIP will be an active or a passive participant

Syntax

```
send-updates [on | off | multiple-interfaces]?
```

Mode

RIP Router Configuration

Parameters

- **on** - configures updates to be sent even if only one interface is present
- **off** - updates will not be sent regardless of the number of interfaces present
- **multiple-interfaces** - optionally specify that, if multiple interfaces (either physical or logical) have addresses on the same subnet, then RIP will send updates only on the first one for which RIP is configured to do so, regardless of configuration

Description

RIP is most often run on a multi-homed box to cause the machine to act as a router in small LANs; however, it is also run on single-homed machines, often to learn the default route or routes to other networks in the domain. RIP will try to guess whether it should act as a router (for example, whether it should act as an active participant in routing) by examining the number of interfaces available to it.
The `send-updates` command can be used to override this default behavior. Specifically, if `send-updates` is specified without the `multiple-interfaces` or `off` options, then RIP updates will be sent even if only one interface is present. This is the same behavior as the `send-updates on` command.

If the `send-updates` command is specified with the `multiple-interfaces` option, and if multiple interfaces have addresses on the same subnet, then RIP will send updates only on the first for which RIP is configured to do so, regardless of the configuration. Finally, if `no send-updates` is specified, or if `send-updates off` is specified, then RIP updates will not be sent, regardless of the number of interfaces present.

The `no send-updates` command is most often used to turn off active participation in a RIP cloud. This can be used, for example, on a multi-homed machine running different protocols on different interfaces where it is desired not to send the other attached networks or learned routes into the RIP cloud. The same functionality can be achieved through policy rules. But for a simple network, that degree of configuration complexity is not necessary.

**Note:** All values of `send-updates` apply only to broadcasting (or multicasting, in the case of RIP version 2) of RIP packets.

**Default**

If `send-updates` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# send-updates multiple-interfaces
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures RIP to send updates, even if only one interface is present.

```
(config)# router rip
(config-router-rip)# send-updates
(config-router-rip)# exit
(config)#
```

**Example 2**

The following example configures RIP to not send any updates to an interface.

```
(config)# router rip
(config-router-rip)# no send-updates
(config-router-rip)# exit
(config)#
```

**source-gateways**

**Name**

`source-gateways` - specifies the routers to which RIP will unicast update messages

**Syntax**

```
source-gateways ipv4_address {n}
no source-gateways ipv4_address
```
Mode
RIP Router Configuration

Parameters

ipv4_address - a valid IPv4 address in dotted-quad notation

{n} - This command can be specified multiple times to configure a complete list of IP addresses to which the configured router will unicast RIP updates.

Description

The source-gateways command allows you to configure a complete list of IP addresses to which the configured router will unicast RIP updates. **Note:** RIP must be configured on an interface that shares a subnet with each gateway address.

If any source gateways are specified, then no RIP packets will be broadcast or multicast. They will only be unicast to the list of source gateways. If no source gateways are specified, Advanced Routing Suite will not unicast updates.

Use the negative form of this command, no source-gateways, with a specific IPv4 address to remove one source gateway address from the configured list. Use the negative form with no specified address to delete the entire list of source gateways.

Default

By default, RIP will either broadcast or multicast its updates. Thus, if source-gateways is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# no source-gateways
```

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

In the following example, a list of three source-gateways is configured. Advanced Routing Suite will unicast updates to the source gateways 192.168.10.1, 101.168.11.1, and 192.168.12.1.

```
(config)# router rip
(config-router-rip)# source-gateways 192.168.10.1
(config-router-rip)# source-gateways 192.168.11.1
(config-router-rip)# source-gateways 192.168.12.1
(config-router-rip)# exit
(config)#
```

Example 2

In the following example, the source-gateway 192.168.11.1 from Example 1 is deleted.

```
(config)# router rip
(config-router-rip)# no source-gateways 192.168.11.1
(config-router-rip)# exit
(config)#
```

Example 3

In the following example, the rest of the entire source-gateways list from Example 1 is deleted. As a result, RIP will send updates by broadcast or multicast.

```
(config)# router rip
```
(config-router-rip)# no source-gateways
(config-router-rip)# exit
(config)#

split-horizon

Name
split-horizon - configures the split horizon feature for RIP

Syntax
split-horizon [ on | off | poison-reverse ]
no split-horizon

Mode
RIP Router Configuration Mode

Parameters
on - RIP will not announce a route back to the interface from which the route is learned
off - RIP will announce a route to all possible interfaces, including the interface from which the route was learned
poison-reverse - RIP will announce a route to all possible interfaces, including the interface from which the route was learned, but with the infinity metric

Description
The split-horizon command configures the way horizon split is performed. This command has three forms. If on is specified, then the simple split-horizon mechanism is enabled. This means that RIP will not announce a route back to the interface from which the route was learned. If off is specified, then the split horizon mechanism is disabled. This means that RIP will announce a route to all possible interfaces, including the interface from which the route was learned. Finally, if poison-reverse is specified, then split horizon with poisoned reverse is enabled. This means that RIP will announce a route to all possible interfaces, including the one from which the route was learned, but with the infinity metric.

The negative form of this command, no split-horizon, removes the configured split horizon type and returns this to its default type.

Default
If split-horizon is not specified, it is the same as if the user had specified the following:

    (config-router-rip)# split-horizon on

Examples

Example 1
In the following example, RIP is configured without split horizon.

    (config)# router rip
    (config-router-rip)# split-horizon off
    (config-router-rip)# exit
    (config)#

Example 2
In the following example, RIP is configured with split horizon with poisoned reverse.
(config)# router rip
(config-router-rip)# split-horizon poison-reverse
(config-router-rip)# exit
(config)#

**Example 3**

In the following example, RIP is configured to return to the default behavior of split horizon. Split horizon will be the effect.

(config)# router rip
(config-router-rip)# no split-horizon
(config-router-rip)# exit
(config)#

term-updates

**Name**

*term-updates* - sets the number of updates sent by RIP during termination

**Syntax**

```
term-updates number
no term-updates number?
```

**Mode**

RIP Router Configuration

**Parameters**

*number* - an integer between 1 and 255, inclusive

**Description**

The `term-updates` command configures the number of updates sent by RIP during termination. The negative form of this command, `no term-updates`, removes the configured *number* value and returns this to its default value of 4. **Note:** Specifying a value for *number* in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `term-updates` is not specified, it is the same as if the user had specified the following:

```
(config-router-rip)# term-updates 4
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a term update number of 6.

```
(config)# router rip
(config-router-rip)# term-updates 6
(config-router-rip)# exit
(config)#
```
timers basic

Name
timers basic – sets the Update Interval, Expiration Time, and Holddown Time for routes received or sent via RIP

Syntax

timers basic update-seconds expiration-seconds holddown-seconds

no timers basic [update-seconds expiration-seconds holddown-seconds]

Mode
RIP Router Configuration

Parameters
update-seconds – the update interval in seconds, specified as an integer from 5 to 4,294,967,295 inclusive
expiration-seconds – the route expiration time in seconds, specified as an integer from 5 to 4,294,967,295 inclusive
holddown-seconds – configures the holddown time in seconds, specified as an integer from 5 to 4,294,967,295 inclusive

Description
The timers basic command sets the Update Interval, the Expiration Time, and the Holddown Time in seconds for routes received and/or sent via RIP. The negative form of this command, no timers basic, returns this value to its default. Specifying values for update-seconds, expiration-seconds, and holddown-seconds in the no form has no effect on the configuration. Thus, they are displayed above as optional.

Note: When configuring timers basic, you must specify an update-seconds value, an expiration-seconds value, and a holddown-seconds value, in that order, even if you want one or two of these to remain the default value.

Default
If timers basic is not configured, it is the same as if the user had specified the following:

(config-router-rip)# timers basic 30 180 120

Command History
NGC 2.2 - This command was introduced.
NGC 2.3 - The Holddown Timer value was added to this command. In addition, each of the previous value ranges changed to 5 to 4,294,967,295 inclusive.

Examples

Example 1
The following example configures the Update Interval to be 25 seconds, the Expiration Time to be 150 seconds, and the Holddown Timer to be 300 seconds.

(config)# router rip
(config-router-rip)# timers basic 25 150 300
(config-router-rip)# exit
(config)#
Example 2
The following example configures the Expiration Time to be 200 seconds. The Update Interval will have the default value of 30 seconds. The Holddown Timer will also have its default value. Note that you must specify the Update Interval and Holddown Timer values even though you are configuring the default.

```
(config)# router rip
(config-router-rip)# timers basic 30 200 120
(config-router-rip)# exit
(config)#
```

Example 3
The following example removes any configured timer values and returns the timers basic command to its default values.

```
(config)# router rip
(config-router-rip)# no timers basic
(config-router-rip)# exit
(config)#
```

trace file

**Name**
trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed

**Syntax**
```
trace file file_name [no-timestamp || overwrite]?
no trace file file_name [no-timestamp || overwrite]?
```

**Mode**
RIP Router Configuration

**Parameters**
- **file_name** - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
- **no-timestamp** - specifies that a timestamp should not be prepended to all trace lines
- **overwrite** - specifies to begin tracing by appending or truncating an existing file

**Description**
The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For RIP, the `trace file` command in RIP Router Configuration Mode specifies a file for tracing of all RIP events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `trace flag` command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands.

**Default**
RIP tracing is turned off by default.
Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, RIP tracing is written to the file "rip.log". No timestamp will display at the beginning of the trace lines.

```
(config)# router rip
(config-router-rip)# trace file rip.log no-timestamp
```

trace flag

Name
```
trace flag - specifies RIP-specific tracing options as well as options that are common across all protocols
```

Syntax
```
trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packets | request | response | other ] [ send | receive | send-receive ]? [ detail? ] )
```
```
no trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packets | request | response | other ] [ send | receive | send-receive ]? [ detail? ] )
```

Mode
```
RIP Router Configuration
```

Parameters
Flags common to all protocols:
```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```
- route – trace routing table changes for routes installed by this protocol or peer
- normal – trace normal protocol occurrences. Note: Abnormal protocol occurrences are always traced.
- state – trace state machine transition in the protocol
- policy – trace the application of protocol and user-specified policy to routes being imported or exported
- task – trace system interface and processing associated with this protocol
- timer – trace timer usage by this protocol
- all – turns on all trace flags
```
IP-specific flags:
```
[ packets | request | response | other ] - These RIP-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```
- packets – trace all RIP packet types
- request – trace RIP information request packets, which include request, poll, and poll entry packets
- response – trace RIP response packets (for example, those that actually contain routing updates)
- other – trace any other type of RIP packet
[send | receive | send-receive ]? - optionally specify whether to limit the tracing to packets sent, received, or both
[detail?] - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

Description
Use the trace flag command to specify tracing flags for RIP tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both task and policy packets in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, trace flags specify that both the sent and received request and response messages are traced in detail. This tracing information will be written to the file rip.log.

(config)# router rip
(config-router-rip)# trace file rip.log
(config-router-rip)# trace flag request send-receive detail
(config-router-rip)# trace flag response send-receive detail
(config-router-rip)# exit
(config)#

trusted-gateways

Name
trusted-gateways - configures the routers from which RIP will accept routes

Syntax
trusted-gateways ipv4-address {n}
no trusted-gateways ipv4-address?

Mode
RIP Router Configuration

Parameters
ipv4-address - a valid IPv4 address in dotted-quad notation
{n} - This command can be repeated an unlimited number of times to created an unlimited number of trusted gateways.

Description
The trusted-gateways command configures routers from which RIP will accept routes. Update messages from routes not specified in this command will be discarded. This allows for additional security beyond that provided by "ip rip authentication" ("ip rip authentication" on page 302). Specifically, it provides a complete list of IP addresses from which the configured router will accept RIP updates. Only those RIP packets originating from the listed hosts will be accepted.
Use the negative form of this command, `no trusted-gateways`, with an IPv4 address to remove a single address from the list of trusted gateways. To delete the entire list, use the negative form of the command with no IPv4 address.

**Default**
All routers are trusted by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example configures RIP to accept updates sent from routes with only host addresses of 2 on the 192.168.10, .12, and .13 networks.

```
(config)# router rip
(config-router-rip)# trusted-gateways 192.168.10.2
(config-router-rip)# trusted-gateways 192.168.11.2
(config-router-rip)# trusted-gateways 192.168.12.2
(config-router-rip)# exit
(config)#
```

**Example 2**
In the following example, the trusted-gateway 192.168.11.1 from Example 1 is deleted.

```
(config)# router rip
(config-router-rip)# no trusted-gateways 192.168.11.1
(config-router-rip)# exit
(config)
```

**Example 3**
In the following example, the rest of the entire trusted-gateways list from Example 1 is deleted. As a result, RIP will trust all routers.

```
(config)# router rip
(config-router-rip)# no trusted-gateways
(config-router-rip)# exit
(config)#
```

**ip rip authentication**

**Name**
`ip rip authentication` - specifies the type of authentication and key values

**Syntax**

```
ip rip authentication [ [simple key] | [md5 id_number md5_key (start-generate date_time || stop-generate date_time || start-accept date_time || stop-accept date_time)?] (0,4) ]
no ip rip authentication
```
Mode

Interface Configuration

Parameters

simple key - specifies simple (clear password) authentication. The value for key is specified by one to eight decimal digits (with a value between 0 and 255) separated by periods, a one- to eight-byte hexadecimal string preceded by 0x, or a one- to eight-character string.

md5 id_number md5_key [(start-generate date_time) || (stop-generate date_time) || (start-accept date_time) || (stop-accept date_time)] - specifies the authentication used for specifying md5 cryptographic authentication. The value for id_number is an integer with a value between 1 and 255, inclusive. The value for md5_key is a one- to sixteen-character string. The start and stop values must be in the format: YYYY-MM-DD.HH.MM. Each start and stop value is optional, and order is not important when specifying multiple commands.

{0,4} - up to 4 MD5 key values can be configured, provided each has a unique md5_key value.

Description

The ip rip authentication command is used both to verify and to generate the authentication field in the RIP header for all packets sent and received on the specified interface. This applies only to version 2 packets and RIPv1-compatible RIPv2 packets, because RIPv1 does not support authentication. One exception to the above is the case of query packets. In the case of querying a router through an interface that is speaking RIPv1, the query packet will still be authenticated, but no authentication will be used on the outgoing packet.

If a packet is received with authentication that does not match, then it is ignored.

On a trusted network, simple authentication can be used to create two logical networks because sets of routers with shared passwords will talk to each other, but not communicate with those using a different password. On an untrusted network, however, this technique should not be used because simple passwords are sent in clear text. MD5 should be used instead. However, even MD5 does not encrypt the entire packet, only the authentication field of the header.

Specifying an authentication without explicitly configure RIP to version 2 will cause a parse error. For somewhat more serious authentication, use trusted gateways or a combination of trusted gateways and authentication.

Default

The default is for no authentication to be explicitly configured.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example configures a simple key authentication for interface fxp1.

```
(config)# interface fxp1
(config-if)# ip rip authentication simple abc
(config-if)# exit
(config)#
```

Example 2

The following example configures md5 authentication with two keys on interface fxp0.

```
(config)# interface fxp0
```
Routing Information
Protocol (RIP)

(config-if)# ip rip authentication md5 1 abc
start-accept 2003-12-01.10.20 start-generate
2003-12-01.10.20

(config-if)# ip rip authentication md5 2 abc
start-accept 2003-12-11.10.20 stop-accept
2003-12-21.10.20

(config-if)# exit
(config)#

ip rip enable

Name
ip rip enable - enables RIP on an interface

Syntax
ip rip enable

Parameters
none

Description
The ip rip enable command explicitly enables RIP on an interface. The no ip rip enable command explicitly disables RIP on an interface.

Default
If ip rip enable is not specified, it is the same as if the user had specified the following:

(config-if)# no ip rip enable

Command History
NGC 2.2 - This command was introduced

Examples
The following example disables RIP on interface fxp2.

(config)# interface fxp2
(config-if)# no ip rip enable
(config-if)# exit
(config)#

ip rip metric-in

Name
ip rip metric-in - sets an additional metric on incoming RIP routers for an interface

Syntax
ip rip metric-in value

no ip rip metric-in value?
Mode
Interface Configuration

Parameters

value - an integer in the range of 0 to 16, inclusive

Description

The `ip rip metric-in` command sets an additional metric on incoming RIP routes for an interface.

It is often the case that a router should prefer routes received on one set of interfaces over those received on another. For example, given two point-to-point links, one can be more expensive than the other and should, therefore, be less preferred. The `ip rip metric-in` command is used for exactly this purpose: to make routes learned from certain interfaces less preferable.

`ip rip metric-in` is the default manner by which RIP increments hop count. That is to say, RIP works by adding a hop every time a route is received and before it is sent back out to other interfaces. This implementation adds the hop when the route is received (for example, before decisions regarding whether the route should be used are made). By default, a metric of 1 plus the kernel interface metric is added as the hop count. Normally, this interface metric is zero, but some operating systems allow it to be specified on interface configuration. If `ip rip metric-in` is explicitly given, it is added as an absolute value (for example, without the interface metric).

The negative form of this command, `no ip rip metric-in`, returns this to its default value. Note: Specifying a `value` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `ip rip metric-in` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip rip metric-in 1
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures a router to add a metric of 2 to the incoming RIP routes on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip rip metric-in 2
(config-if)# exit
(config)#
```

**ip rip metric-out**

Name

`ip rip metric-out` - sets an additional metric on outgoing RIP routes for an interface

Syntax

```
ip rip metric-out value
no ip rip metric-out value?
```

Mode

Interface Configuration
Parameters

value – an integer between 0 and 16, inclusive

Description

The `ip rip metric-out` command configures an addition metric on outgoing RIP routes for an interface. Normally, this RIP implementation adds to the hop count only on incoming routes. There are times, however, when the user wants to cause other routers not to prefer routes from a given origin. For example, if the router is a backup router, it might be desirable for its routes to always be less preferred. `ip rip metric-out` accomplishes this by adding to the RIP metric on top of any metric specified by `ip rip metric-in` before RIP updates are sent out the specified interface.

Use the negative form of this command, `no ip rip metric-out`, to remove a configured value and return this to its default value of 0. **Note:** Specifying a value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip rip metric-out` is not specified, it is the same as if the user had specified the following:

```
(config-if) ip rip metric-out 0
```

Example

The following example configures a router to add a cost of 2 to outgoing RIP routes for interface fxp0.

```
(config)# interface fxp0
(config-if)# ip rip metric-out 2
(config-if)# exit
(config)#
```

**ip rip no-receive**

Name

`ip rip no-receive` - specifies whether RIP will listen to or discard RIP updates

Syntax

```
ip rip no-receive
no ip rip no-receive
```

Mode

Interface Configuration

Parameters

none

Description

The `ip rip no-receive` command specifies whether RIP will listen to RIP updates received on a given interface. Although it would almost certainly be a mis-configuration, it is important to note that RIP can send RIP updates on a superset of those interfaces on which it receives updates. This can be a valid configuration if, for example, the user receives RIP updates from an ISP, then redistributes those onto the LAN, and does not want to send the LAN topology back to the ISP.
Default
By default, RIP listens to updates received on a given interface. Therefore, if `ip rip no-receive` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip rip no-receive
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures no-receive on interface fxp0. As a result, updates received on interface fxp0 will be discarded.

```
(config)# interface fxp0
(config-if)# ip rip no-receive
(config-if)# exit
(config)#
```

ip rip no-send

Name
`ip rip no-send` - specifies whether RIP will send RIP updates

Syntax
```
ip rip no-send
no ip rip no-send
```

Mode
Interface Configuration

Parameters
none

Description
RIP, by default, sends updates on interfaces on which RIP is running. The `ip rip no-send` command specifies that RIP will not send updates.

Default
If `ip rip no-send` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip rip no-send
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the no-send option on interface fxp1. As a result, no updates will be sent over that interface.

```
(config)# interface fxp1
(config-if)# ip rip no-send
(config-if)# exit
```
ip rip secondary-authentication

**Name**

ip rip secondary-authentication - sets the authentication used on an interface if the primary authentication fails

**Syntax**

```plaintext
ip rip secondary-authentication [ [simple key] | [md5 id_number md5_key (start-generate date_time || stop-generate date_time || start-accept date_time || stop-accept date_time)?] {0,4} ]

no ip rip secondary-authentication
```

**Mode**

Interface Configuration

**Parameters**

- **simple key** - specifies simple (clear password) authentication. The value for `key` is specified by one to eight decimal digits (with a value between 0 and 255) separated by periods, a one- to eight-byte hexadecimal string preceded by 0x, or a one- to eight-character string.

- **md5 id_number md5_key (start-generate date_time || stop-generate date_time || start-accept date_time || stop-accept date_time)?** - specifies the authentication used for specifying md5 cryptographic authentication. The value for `id_number` is an integer with a value between 1 and 255, inclusive. The value for `md5_key` is specified by one to eight decimal digits (with a value between 0 and 255) separated by periods, a one- to eight-byte hexadecimal string preceded by 0x, or a one- to eight-character string. The `start` and `stop` values must be in the format: YYYY/MM/DD HH:MM.

- `{0,4}` - up to 4 md5 key values can be configured, provided each has a unique `md5_key` value.

**Description**

The `ip rip secondary-authentication` is identical in function to the primary authentication configured using the `ip rip authentication` command but is only used if the primary authentication fails. This type of authentication can be used while a network is in transition to verify that old passwords are still accepted.

Use the negative form of this command, `no ip rip secondary-authentication`, to remove the configured authentication.

**Default**

The default is for no authentication to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures a simple key authentication for interface fxp0.

```plaintext
(config)# interface fxp0
(config-if)# ip rip secondary-authentication simple abc
(config-if)# exit
```
Example 2

The following example configures md5 authentication with two keys on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip rip secondary-authentication md5 1 abc
             start-accept 2003/12/01 10:20 start-generate 2003/12/01 10:20
(config-if)# ip rip secondary-authentication md5 2 abc
             start-accept 2003/12/11 10:20 stop-accept 2003/12/21 10:20
(config-if)# exit
(config)#
```

**ip rip version**

**Name**

*ip rip version* - specifies the version of RIP to be run on an interface

**Syntax**

```
ip rip version [ 1 | 2 [ broadcast | multicast ]? ]
no ip rip version
```

**Mode**

Interface Configuration

**Parameters**

1 - specify to run RIP version 1 on the interface
2 [ broadcast | multicast ] - specify to run RIP version 2 on the interface. Also, optionally specify whether updates will be sent to a multicast group or will be broadcast to the network

**Description**

The *ip rip version* command specifies the version of RIP to be run. This command is used to override the default version of RIP that will be run on a given interface. Normally, RIPv1 will be run on all interfaces. If version 2 is specified, then the default behavior depends on the capabilities of the interface. If the interface is multicast capable, then RIP updates will be multicast to RIP2-ROUTERS.MCAST.NET (the reserved multicast address, 224.0.0.9). If the interface is not multicast capable, then RIP version 1 compatible version 2 will be broadcast.

The optional broadcast and multicast parameters are effective only if RIP version 2 is configured. Specifying broadcast or multicast allows the above default behavior to be overridden. This is normally used to specify that only version-1-compatible packets should be sent for interoperability purposes, even though a given interface is multicast capable. An exception is the case in which version 2 multicast is specified on an interface that is not multicast capable (for example, a point-to-point link). In this case, if a source-gateway is also specified, then the full RIPv2 packets will be directly unicast to the source gateway on the specified interface.

**Default**

If *ip rip version* is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip rip version 1
```

If version 2 is being configured on multicast-capable interfaces without specifying multicast or broadcast, it is the same as if the user had specified the following:

```
(config-if)# ip rip version2 multicast
```
Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures version 2 broadcast on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip rip version 2 broadcast
(config-if)# exit
(config)#
```

Example 2
In the following example, the version setting in Example 1 is set back to using the default behavior for interface fxp0. Version 1 packets will then be broadcast to the network.

```
(config)# interface fxp0
(config-if)# no ip rip version
(config-if)# exit
(config)#
```

Example 3
The following example configures version 2 on multicast-capable interface fxp1. Updates will be sent out to multicast group 224.0.0.9.

```
(config)# interface fxp1
(config-if)# ip rip version 2
(config-if)# exit
(config)#
```

show ip rip database

Name
```
show ip rip database - displays information about a specific route or all routes in the Routing Information Base
```

Syntax
```
show ip rip database [(ipv4-address mask) | (tag value)]?
```

Mode
User Execution

Parameters
```
ipv4-address mask - optionally enter a specific IPv4 address and subnet mask argument about which routing information should be displayed
tag value - optionally specify a tag and value used to mark a specific route. The tag value can be an integer from 0 to 4,294,967,295, inclusive.
```
Description
The `show ip rip database` query displays information about routes in the Routing Information Base. This query has three forms. If the query is issued without arguments, then information about all RIP routes is returned. Alternatively, the query can be issued with a specific IPv4 address and mask. In this case, the reply will contain information pertaining only to the referenced address. Finally, a query can be submitted with a tag value. In this case, all RIP routes that match the tag will be displayed.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example displays all rip routes.

```plaintext
> show ip rip database
192.168.11.0/24 directly connected, fxp0
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, fxp0
[2] via 192.168.15.2, 00:00:20, fxp1
182.168.13.0/24
[1] via 182.168.14.2, 00:00:25, fxp3
```

Example 2
In the following example, a query is submitted for RIP route information for a specific network.

```plaintext
> show ip rip database 192.168.13.0 255.255.255.0
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, fxp0
[2] via 192.168.15.2, 00:00:20, fxp1
```

Example 3
In the following example, a query is submitted for information about RIP routes that match a given tag, 1000. The return shows one route that matches the tag.

```plaintext
> show ip rip database tag 1000
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, fxp3
```

Field Descriptions
The following table describes the fields that appear in the RIP Database Query (using fields in Example 1).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.11.0/24</td>
<td>Destination IPv4 address and mask length</td>
</tr>
<tr>
<td>directly connected</td>
<td>Displays if the network is directly connected</td>
</tr>
<tr>
<td>[1] via 192.168.14.2</td>
<td>The first gateway</td>
</tr>
<tr>
<td>[2] via 192.168.15.2</td>
<td>The second gateway</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>00:00:25</td>
<td>Last heard time in hours:minutes:seconds</td>
</tr>
<tr>
<td>fxp0</td>
<td>The interface from which a nexthop can be reached</td>
</tr>
</tbody>
</table>
Chapter 14

SNMP Multiplexing (SMUX)

In This Chapter

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP Multiplexing (SMUX) Overview</td>
<td>313</td>
</tr>
<tr>
<td>smux password</td>
<td>313</td>
</tr>
<tr>
<td>smux port</td>
<td>314</td>
</tr>
<tr>
<td>smux trace file</td>
<td>315</td>
</tr>
<tr>
<td>smux trace flag</td>
<td>315</td>
</tr>
</tbody>
</table>

SNMP Multiplexing (SMUX) Overview

Support for the SMUX (RFC 1227) protocol is unique to Advanced Routing Suite due to its intended installation as a daemon running on a multi-user timeshare operating system. While this paradigm continues to evolve, an original intent of the SMUX protocol implement was to allow Advanced Routing Suite to interoperate with another daemon handling the SNMP protocol and communicate with management stations. This allows Advanced Routing Suite to handle only a set of SNMP subtrees relating to the protocols implemented there.

Advanced Routing Suite supports SMUX as the only way to interact with the MIB modules. Only SNMP version 1 is supported, and all MIB variables are read-only.

Upon contacting the master agent, a string password and SNMP Object Identifier identity are passed for authentication purposes. If the authentication succeeds, Advanced Routing Suite will register the routing MIB subtrees and request that it be contacted when the master agent receives queries for these subtrees. When the master agent receives such a query from a management station, it will be passed to Advanced Routing Suite.

smux password

Name

smux password - specifies the clear text password to be used for authentication

Syntax

smux password string_value

no smux password string_value?

Mode

Global Configuration

Parameters

string_value - a string with a maximum of 255 characters for the SMUX opening password

Description

The SMUX protocol allows a clear text password to be used for authentication of the subagent. This password is configured using the smux password command. The negative of this command, no smux
password, removes the specified string_value and returns to the default of a blank password. **Note:** Specifying a value for string_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
The default is a blank password with 0 length.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures "abc" as the clear text password for authentication of the subagent

```
(config)# smux password abc
```

### smux port

**Name**
smux port - specifies the port on which to contact the master agent via TCP

**Syntax**

```
smux port port_value
no smux port port_value?
```

**Mode**
Global Configuration

**Default**

port_value - a port number on which to contact the master agent. This must be an integer from 0 to 65535, inclusive.

**Description**
The master agent can be running on a different port than the well-known port of 199. This allows Advanced Routing Suite to contact the agent on the specified port. The negative of this command, no smux port, removes the specified port_value and returns this to the default port value of 199. **Note:** Specifying a value for port_value in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If smux port is not specified, it is the same as if the user had specified the following:

```
(config)# smux port 199
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
In the following example, the master agent is configured to run on port 2112.

```
(config)# smux port 2112
```
**smux trace file**

**Name**

*smux trace file* - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed when tracing in SNMP Multiplexing

**Syntax**

```plaintext
smux trace file file_name [no-timestamp || overwrite]?
no smux trace file file_name [no-timestamp || overwrite]?
```

**Mode**

Global Configuration

**Parameters**

- `file_name` - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
- `no-timestamp` - specifies that a timestamp should not be prepended to all trace lines
- `overwrite` - specifies to begin tracing by appending or truncating an existing file

**Description**

The *trace file* command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For SMUX, the *smux trace file* command specifies a file for tracing of all SMUX events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the *smux trace flag* command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands. Consider the following example:

```plaintext
(config)# smux trace file smux.log no-timestamp
```

**Default**

SMUX tracing is turned off by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, SMUX tracing is written to the file "smux.log". No timestamp will display at the beginning of the trace lines.

```plaintext
(config)# smux trace file smux.log no-timestamp
```

**smux trace flag**

**Name**

*smux trace flag* - specifies SMUX-specific tracing options as well as options that are common across all protocols
Syntax

```
smux trace flag ( [ route | normal | state | policy | task | timer | all | send | receive | packets ] )
no smux trace flag ( [ route | normal | state | policy | task | timer | all | send | receive | packets ] )
```

Mode
Global Configuration

Parameters

```
[ route | normal | state | policy | task | timer | all | send | receive | packets ]
```
- These tracing flags are common to all protocols. They cannot be associated with a send, receive, or detail action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
  - **route** - trace routing table changes for routes installed by this protocol or peer
  - **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
  - **state** - trace state machine transition in the protocol
  - **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
  - **task** - trace system interface and processing associated with this protocol
  - **timer** - trace timer usage by this protocol
  - **all** - turns on all trace flags
  - **send** - trace packets sent by the master agent
  - **receive** - trace packets received from the master agent
  - **packets** - trace every byte of traced packet

Description
Use the `smux trace flag` command to specify tracing flags for SMUX tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both packets and send in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, flags specify to trace packets and receive information. This tracing information will be written to the file `smux.log`.

```
(config)# smux trace file smux.log max-size 1024k
(config)# smux trace flag packets
(config)# smux trace flag receive
```
Chapter 15

Distance Vector Multicast Routing Protocol (DVMRP)

In This Chapter

Distance Vector Multicast Routing Protocol (DVMRP) Overview 317
ip dvmrp 317
ip dvmrp distance 318
ip dvmrp default-metric 319
ip dvmrp disable 320
ip dvmrp metric-offset 320
ip dvmrp nodvmrpout 321
ip dvmrp noretransmit 322
ip dvmrp prune-lifetime 323
ip dvmrp trace file 323
ip dvmrp trace flag 324
ip dvmrp unicast-routing 326
show ip dvmrp interfaces 327
show ip dvmrp neighbors 328
show ip dvmrp route 329
tunnel mode dvmrp 331

Distance Vector Multicast Routing Protocol (DVMRP) Overview

DVMRP is the original IP multicast routing protocol. It was designed to run over both multicast capable LANs (like Ethernet) as well as through non-multicast capable routers. In the case of non-multicast capable routers, the IP multicast packets are "tunneled" through the routers as unicast packets. Because DVMRP replicates the packets, it has an effect on performance, but has provided an intermediate solution for IP multicast routing on the Internet while router vendors decide to support native IP multicast routing.

When configured, DVMRP defaults to enabling all interfaces that are multicast capable.

ip dvmrp

Name
ip dvmrp - enables the DVMRP protocol

Syntax

    ip dvmrp
    no ip dvmrp
**Mode**  
Interface Configuration

**Parameters**  
none

**Description**  
The `ip dvmrp` command enables the DVMRP protocol on the referenced interface. This command turns on both DVMRP tree-building operations and DVMRP routing. The `no ip dvmrp` command turns off both DVMRP tree building and routing.

**Default**  
By default, DVMRP is not enabled on an interface.

**Command History**  
NGC 2.2 - This command was introduced.

**Examples**  
The following example enables DVMRP on interfaces eth0 and eth1.

```
(config)# interface eth0
(config-if)# ip dvmrp
(config-if)# exit
(config)# interface eth1
(config-if)# ip dvmrp
(config-if)# exit
(config)#
```

**ip dvmrp distance**

**Name**  
`ip dvmrp distance` – configures the value that Advanced Routing Suite uses for DVMRP routes in the active route selection process

**Syntax**  
```
  ip dvmrp distance pref
  no ip dvmrp distance pref?
```

**Mode**  
Global Configuration

**Parameters**  
`pref` – an integer between 0 and 255, inclusive

**Description**  
The `ip dvmrp distance` command specifies how active routes that are learned from DVMRP will be selected, compared to other protocols. When a route has been learned from more than one protocol, the active route will be selected from the protocol with the lowest distance. This value applies to all interfaces on which DVMRP is running.
The negative form of this command, `no ip dvmrp distance`, removes the configured value and returns this to its default value of 70. **Note:** Specifying a value for `pref` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip dvmrp distance` is not specified, it is the same as if the user had specified the following:

```
(config)# ip dvmrp distance 70
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the administrative distance for DVMRP to be 40.

```
(config)# ip dvmrp distance 40
```

### ip dvmrp default-metric

**Name**

`ip dvmrp default-metric` - specifies the default value for the interface metric

**Syntax**

```
ip dvmrp default-metric value  
nm ip dvmrp default-metric value?
```

**Mode**

Global Configuration

**Parameters**

`value` - an integer ranging from 1 to 32, inclusive

**Description**

The `ip dvmrp default-metric` command specifies the default metric applied to all interfaces on which DVMRP is running. This value can be overridden at the interface level. (See "ip dvmrp metric-offset" ("ip dvmrp metric-offset" on page 320).)

The negative form of this command, `no ip dvmrp default-metric`, removes the configured value and returns this to its default value of 1. **Note:** Specifying a value for `value` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip dvmrp default-metric` is not specified, it is the same as if the user had specified the following:

```
(config)# ip dvmrp default-metric 1
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures a default metric of 10.

```
(config)# ip dvmrp default-metric 10
```
ip dvmrp disable

Name
ip dvmrp disable - disables DVMRP on an interface

Syntax
   ip dvmrp disable
   no ip dvmrp disable

Mode
Interface Configuration

Parameters
none

Description
The ip dvmrp disable command specifies that DVMRP messages received on the current interface must be ignored. The negative form of this command, no ip dvmrp disable, re-enables DVMRP on an interface.

Default
DVMRP runs on all interfaces by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example disables DVMRP on interface fxp1.
   (config)# interface fxp1
   (config-if)# ip dvmrp disable
   (config-if)# exit
   (config)#

ip dvmrp metric-offset

Name
ip dvmrp metric-offset - specifies the default value for the interface metric

Syntax
   ip dvmrp metric-offset in metric
   no ip dvmrp metric-offset in metric?

Mode
Interface Configuration

Parameters
   in - specify that this metric applies to the incoming interface
   metric - an integer value ranging from 1 to 32, inclusive
Description
The `ip dvmrp metric-offet` command sets the metric to add to all routes learned from neighbors reached through this interface. The negative form of this command, `no ip dvmrp metric-offet`, removes the configured value and re-sets this to its default value of 1. **Note:** Specifying a value for `metric` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `ip dvmrp metric-offet` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip dvmrp metric-offet in 1
```

Command History
NGC 2.2 - This command was introduced.

Examples
Example 1
In the following example, a metric of 5 is added to the route metric for each route learned from neighbors reached through interface eth0.

```
(config)# interface eth0
(config-if)# ip dvmrp metric-offet in 5
(config-if)# exit
(config)#
```

Example 2
In the following example, a metric of 8 is added to the route metric for each route learned from neighbors reached through interface eth1.

```
(config)# interface eth1
(config-if)# ip dvmrp metric-offet in 8
(config-if)# exit
(config)#
```

`ip dvmrp nodvmrpout`

Name
`ip dvmrp nodvmrpout` - tells DVMRP to only listen on an interface

Syntax
```
ip dvmrp nodvmrpout
no ip dvmrp nodvmrpout
```

Mode
Interface Configuration

Parameters
none

Description
The `ip dvmrp nodvmrpout` command disables DVMRP as a speaker on an interface, although it will continue to listen and accept routes.
Default
By default, DVMRP acts as a speaker. Therefore, if `ip dvmrp nodvmrpout` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip dvmrp nodvmrpout
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example causes interface eth0 to listen and accept routes, but not to act as a speaker.

```
(config)# interface eth0
(config-if)# ip dvmrp nodvmrpout
(config-if)# exit
(config)#
```

`ip dvmrp noretransmit`

Name
`ip dvmrp noretransmit` - specifies to refrain from re-sending DVMRP prune packets

Syntax
```
ip dvmrp noretransmit
no ip dvmrp noretransmit
```

Mode
Interface Configuration

Parameters
none

Description
The `ip dvmrp noretransmit` command configures Advanced Routing Suite to not perform the exponential backoff prune retransmission. After the transmission of the first prune, no additional prunes will be transmitted on reception of data until the prune lifetime has expired.

Default
The exponential backoff mechanism is performed by default. Therefore, if `ip dvmrp noretransmit` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip dvmrp noretransmit
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures interface eth1 to not perform the exponential prune backoff mechanism.

```
(config)# interface eth1
(config-if)# ip dvmrp noretransmit
(config-if)# exit
```
ip dvmrp prune-lifetime

Name
ip dvmrp prune-lifetime - specifies the maximum default lifetime of prunes in seconds

Syntax
   ip dvmrp prune-lifetime time-seconds
   no ip dvmrp prune-lifetime time-seconds?

Mode
Global Configuration

Parameters
   time-seconds - an integer value representing a number of seconds ranging from 0 to 2,147,483,648, inclusive

Description
The ip dvmrp prune-lifetime command configures the maximum value to be placed into a prune message. The actual lifetime value is the minimum of all the downstream prunes for the source and a randomized value that falls between one-half the prune lifetime and the prune lifetime. The value is in seconds.

The negative form of this command, no dvmrp prune-lifetime, removes the configured time-seconds value and returns this to its default value of 7200 seconds for neighbors that support Generation ID and 300 seconds for neighbors that do not support Generation ID.

Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If ip dvmrp prune-lifetime is not specified, it is the same as if the user had specified the following for neighbors that support Generation ID:

   (config)# ip dvmrp prune-lifetime 7200

If ip dvmrp prune-lifetime is not specified, it is the same as if the user had specified the following for neighbors that do not support Generation ID:

   (config)# ip dvmrp prune-lifetime 300

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the prune lifetime to be 7000 seconds.

   (config)# ip dvmrp prune-lifetime 7000

ip dvmrp trace file

Name
ip dvmrp trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed
Distance Vector Multicast Routing Protocol (DVMRP)

Syntax

```
ip dvmrp trace file file_name [no-timestamp || overwrite]?
no ip dvmrp trace [no-timestamp || overwrite]?
```

Mode

Global Configuration

Parameters

file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.

no-timestamp - specifies that a timestamp should not be prepended to all trace lines

overwrite - specifies to begin tracing by appending or truncating an existing file

Description

The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For DVMRP, the `ip dvmrp trace file` command specifies a file for tracing of all DVMRP events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `ip dvmrp trace flag` command.

The `no-timestamp` option disables the prepending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands. Consider the following example:

```
(config)# ip dvmrp trace file /var/log/dvmrp.log no-timestamp
(config)# ip dvmrp trace file /var/log/dvmrp.log
```

The option given in the second command completely replaces that given in the first. In order to specify both `no-timestamp` and `max-files 10`, they must be entered on the same line as follows.

```
(config)# ip dvmrp trace file /var/log/dvmrp.log no-timestamp
```

Default

DVMRP tracing is turned off by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, DVMRP tracing is written to the file "/var/tmp/dvmrp.log". No timestamp will display at the beginning of the trace lines.

```
(config)# ip dvmrp trace file /var/tmp/dvmrp.log no-timestamp
```

---

**ip dvmrp trace flag**

Name

```
ip dvmrp trace flag - specifies DVMRP-specific tracing options as well as options that are common across all protocols
```

Syntax

```
ip dvmrp trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packets | probe |
```
```
no ip dvmrp trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packets | probe | report | prune | graft ] [ send | receive | send-receive ]? [detail?] )
```

**Mode**

Global Configuration

**Parameters**

Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. Note: Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

DVMRP-specific flags:

```
[ packets | probe | report | prune | graft ] - These DVMRP-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```

- **packets** - trace all DVMRP packet types
- **probe** - trace DVMRP probe packets
- **report** - trace DVMRP report packets
- **prune** - trace DVMRP prune packets
- **graft** - trace DVMRP graft packets

```
[send | receive | send-receive ]? - optionally specify whether to limit the tracing to packets sent, received, or both
```

[detail?] - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

**Description**

Use the `ip dvmrp trace flag` command to specify tracing flags for DVMRP tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both probe and prune packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.
Examples
In the following example, trace flags specify that both the sent and received prune and graft messages are traced in detail. This tracing information will be written to the file /var/tmp/dvmrp.log.

(config)# ip dvmrp trace file /var/tmp/dvmrp.log
(config)# ip dvmrp trace flag graft send-receive detail
(config)# ip dvmrp trace flag prune send-receive detail

**ip dvmrp unicast-routing**

**Name**
ip dvmrp unicast-routing - configures the interfaces to perform only DVMRP route exchange

**Syntax**

```
ip dvmrp unicast-routing
no ip dvmrp unicast-routing
```

**Mode**
Interface Configuration

**Parameters**
none

**Description**
The `ip dvmrp unicast-routing` command configures the specified interfaces to perform only DVMRP route exchange. This is different than the "ip dvmrp" ("ip dvmrp" on page 317) command, which enables both tree-building operations and routing. When `ip dvmrp unicast-routing` is configured, the DVMRP multicast delivery tree-building operations will not be performed on the interface. The negative form of this command, `no ip dvmrp unicast-routing`, effectively disables DVMRP, as DVMRP tree-building operations require DVMRP routing functionality.

**Default**
Multicast delivery tree-building operations are performed by default. Therefore, if `ip dvmrp unicast-routing` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip-dvmrp unicast-routing
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
This command turns on DVMRP routing.

```
(config)# interface fxp0
(config-if)# ip dvmrp unicast-routing
(config-if)# exit
(config)#
```
Example 2

The following example turns on DVMRP routing (no tree building) on the interface. Note what the `ip dvmrp` command does: `ip dvmrp` turns on both DVMRP tree-building and routing. `no ip dvmrp` will turn off BOTH DVMRP tree-building and DVMRP routing.

The following sequence of commands can be used to go from BOTH tree-building and routing to just routing.

```
(config)# interface fxp0
(config-if)# ip dvmrp
(config-if)# end
```

Both DVMRP tree-building and routing are on as a result of the above commands. Now to migrate to just routing.

```
(config)# interface fxp0
(config-if)# no ip dvmrp
(config-if)# ip dvmrp unicast-routing
(config-if)# end
```

**show ip dvmrp interfaces**

**Name**

`show ip dvmrp interfaces` - displays information about the interfaces on which DVMRP is enabled

**Syntax**

```
show ip dvmrp interfaces
```

**Mode**

User Execution

**Parameters**

none

**Description**

The `show ip dvmrp interfaces` query displays information about all interfaces on which DVMRP is running.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example returns information about all interfaces on which DVMRP is enabled.

```
> show ip dvmrp interfaces
  eth1, 192.168.10.1, vif 1, with 3 DVMRP neighbors
  5 bad DVMRP packets received, 2 bad DVMRP routes received
  eth2, 192.168.100.1, vif 2, with 1 DVMRP neighbor
  0 bad DVMRP packets received, 0 bad DVMRP routes received
```

**Field Descriptions**

The following table describes the fields that appear in the DVMRP Interfaces Query.
Distance Vector Multicast Routing Protocol (DVMRP)

Table 20-1  DVMRP Interfaces Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1</td>
<td>The physical interface name</td>
</tr>
<tr>
<td>192.168.10.1</td>
<td>The IPv4 address used on the interface by the DVMRP protocol</td>
</tr>
<tr>
<td>vif</td>
<td>The Vif number used to represent this interface to the kernel</td>
</tr>
<tr>
<td>DVMRP neighbors</td>
<td>The number of DVMRP neighbors reachable on the interface</td>
</tr>
<tr>
<td>packets</td>
<td>The number of bad DVMRP packets received on the interface</td>
</tr>
<tr>
<td>routes</td>
<td>The number of bad DVMRP routes received on the interface</td>
</tr>
</tbody>
</table>

show ip dvmrp neighbors

Name
show ip dvmrp neighbors - displays information about this router’s DVMRP neighbors

Syntax
show ip dvmrp neighbors

Mode
User Execution

Parameters
none

Description
The show ip dvmrp neighbors query displays information about all of this router’s DVMRP neighbors.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example shows information returned for all DVMRP neighbors.

> show ip dvmrp neighbors
  192.168.10.1 uptime 01:10:55, expires 00:00:25
  via eth0, [version 3.255] [flags: GPM]
  192.168.10.2 uptime 01:10:52, expires 00:00:21
  via eth0, [version 3.255] [flags: GPM]
  192.168.100.1 uptime 02:01:25, expires 00:00:19
  via eth1, [version 3.255] [flags: GPM]
Field Descriptions
The following table describes the fields that appear in the DVMRP Neighbors Query.

Table 20-1 DVMRP Neighbors Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.1</td>
<td>The DVMRP neighbor address</td>
</tr>
<tr>
<td>uptime</td>
<td>Length of time the neighbor has been known. Expressed in hours:minutes:seconds.</td>
</tr>
<tr>
<td>expires</td>
<td>The length of time until the neighbor will be removed unless refreshed. Expressed in hours:minutes:seconds.</td>
</tr>
<tr>
<td>via</td>
<td>The interface through which the neighbor can be reached</td>
</tr>
<tr>
<td>version</td>
<td>The DVMRP version of the neighbor router</td>
</tr>
<tr>
<td>flags</td>
<td>The capabilities of the neighbor router</td>
</tr>
<tr>
<td></td>
<td>G: Neighbor supports GenID</td>
</tr>
<tr>
<td></td>
<td>P: Neighbor supports prune mechanism</td>
</tr>
<tr>
<td></td>
<td>M: Neighbors supports mtrace</td>
</tr>
</tbody>
</table>

show ip dvmrp route

Name
show ip dvmrp route - displays information about the routes in the DVMRP routing table.

Syntax

    show ip dvmrp route [ipv4-address | interface-name]?

Mode
User Execution

Parameters
ipv4-address - optionally specify a valid IPv4 address in dotted-quad notation
interface-name - optionally specify a physical interface name

Description
The show ip dvmrp route query displays information about the routes in the DVMRP routing table. When no parameter is given, then information about all DVMRP routes is displayed. When an IPv4 address
is given, then information about the best match route is displayed. When a physical interface name is given, then information about only those reached through that interface are displayed.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example returns information about all DVMRP routes.

```
> show ip dvmrp route
10.5.0.0/16 [70/10] uptime 00:5:45, expires 00:01:10
via 192.168.20.1, fxp3
192.168.10.0/25 [70/5] uptime 01:23:55, expires 00:02:24
via 172.0.0.10, fxp0
192.168.10.1/25 [70/5] uptime 01:23:55, expires 00:02:24
via 172.0.0.20, fxp0
```

**Example 2**

The following example returns information for the best match routes to 192.168.10.1.

```
> show ip dvmrp route 192.168.10.1
DVMRP Routing Table - 1 Entry
192.168.10.0/25 [70/5] uptime 01:23:55, expires 00:02:24
via 172.0.0.10, fxp0
```

**Example 3**

The following example returns information about destinations reached through interface fxp0.

```
> show ip dvmrp route fxp0
DVMRP Routing Table - 1 Entry
192.168.10.0/25 [70/5] uptime 01:23:55, expires 00:02:24
via 172.0.0.10, fxp0
192.168.10.1/25 [70/5] uptime 01:23:55, expires 00:02:24
via 172.0.0.20, fxp0
```

**Field Descriptions**

The following table describes the that appear in the DVMRP Route to Source Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.0/25</td>
<td>The source network</td>
</tr>
<tr>
<td>[70/5]</td>
<td>Preference/metric of a route</td>
</tr>
<tr>
<td>uptime</td>
<td>Length of time the route has been in the routing table. This is expressed in hours:minutes:seconds</td>
</tr>
<tr>
<td>expires</td>
<td>Length of time until the route will be removed, unless refreshed. This is expressed in hours:minutes:seconds</td>
</tr>
</tbody>
</table>
### tunnel mode dvmrp

**Name**

tunnel mode dvmrp - configures Advanced Routing Suite to perform old-style DVMRP tunnel encapsulation

**Syntax**

```
tunnel mode dvmrp
no tunnel mode dvmrp
```

**Mode**

Interface Configuration

**Parameters**

none

**Description**

The `tunnel mode dvmrp` command configures Advanced Routing Suite to perform old-style DVMRP tunnel encapsulation. In old-style tunnel encapsulation, DVMRP control messages are non ip-ip encapsulated, but merely unicasted to the tunnel endpoint.

**Default**

If `tunnel mode dvmrp` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no tunnel mode dvmrp
```

**NGC 2.2** - This command was introduced.

**Examples**

In the following example, a tunnel is configured on logical interface 192.168.15.1 with a remote end point of 192.168.10.1. DVMRP tunnel mode is then enabled on the tunnel.

```
(config)# ipip-tunnel 192.168.15.1 192.168.10.1
(config)# interface 192.168.10.1
(config-if)# tunnel mode dvmrp
(config-if)# exit
(config)#
```
Chapter 16

Internet Group Management Protocol (IGMP)

In This Chapter

Internet Group Management Protocol (IGMP) Overview 332
clear ip igmp group 333
ip igmp 334
ip igmp ignore-v1-messages 335
ip igmp ignore-v2-messages 335
ip igmp last-member-query-count 336
ip igmp last-member-query-interval 337
ip igmp query-interval 339
ip igmp query-max-response-time 340
ip igmp require-router-alert 341
ip igmp robustness 342
ip igmp send-router-alert 343
ip igmp startup-query-count 344
ip igmp startup-query-interval 345
ip igmp static-group 347
ip igmp trace file 348
ip igmp trace flag 349
ip igmp version 350
show ip igmp groups 351
show ip igmp interface 355
show ip igmp interface-summary 358
show ip igmp static-groups 359

Internet Group Management Protocol (IGMP) Overview

IGMP was designed for hosts on multi-access networks to inform locally-attached routers of their multicast group memberships. Hosts inform routers of the groups of which they are members by multicasting IGMP Group Membership Reports. Once multicast routers listen for these reports, they can exchange group membership information with other multicast routers. This reporting system allows distribution trees to be formed to deliver multicast datagrams. The original version of IGMP was defined in RFC 1112, Host Extensions for IP Multicasting. Extensions to IGMP, known as IGMP version 2, include explicit Leave messages for faster pruning and are defined in RFC 2236. Advanced Routing Suite implements IGMP version 2, which includes interoperability with version 1 hosts, and version 3, which includes interoperability with version 2 and version 1 hosts. The original version of IGMP can be found at:

http://www.ietf.org/rfc/rfc1112.txt

IGMP version 2 is described in:
http://www.ietf.org/rfc/rfc2236.txt

IGMP version 3 is described in:
clear ip igmp group

Name

clear ip igmp group - removes IGMP join state

Syntax

clear ip igmp group [ group-name || group-address || type || number ]?

Mode

Privileged Execution

Parameters

group-name - the name of the multicast group, as defined in DNS host table. Entering this is optional.
group-address - the IPv4 address of the multicast group in dotted-quad format. Entering this is optional.
type - the interface type. Entering this is optional.
number - the interface number. Entering this is optional.

Description

Use clear ip igmp group to remove IGMP join state. This command has four forms, depending upon whether any or all of the optional parameters are provided. If only the group name or group address is specified, then the associated group entries will be deleted from all interfaces. If only the interface is specified, then all group entries will be deleted for the interface. If neither a group nor an interface is specified, then group memberships over all the IGMP interfaces will be deleted.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example shows a request to clear group entry for 224.1.1.1 on interface 192.168.10.1.

    # clear ip igmp group 224.1.1.1 192.168.10.1

Example 2

The following example shows a request to clear group entries for interface Ethernet0/1.

    # clear ip igmp group Ethernet0/1

Example 3

The following example shows a request to clear group entries for group 224.1.1.1 on interface Ethernet0/1.

    # clear ip igmp group 224.1.1.1 Ethernet0/1

Example 4

The following example, a request is issued to clear the same group entries for group 224.1.1.1 on interface Ethernet0/1, as in Example 3. The difference is that in this example, the interface is specified first, followed by the group.

    # clear ip igmp group Ethernet0/1 224.1.1.1
Example 5
The following example issues a request to clear group entry without any parameters. This essentially deletes all group entries over all interfaces.

    # clear ip igmp group

ip igmp

Name
ip igmp - enables or disables the IGMP protocol

Syntax
ip igmp
no ip igmp

Mode
Interface Configuration

Parameters
none

Description
The `ip igmp` configuration enables the IGMP protocol on the associated interface with the default set of configurations. Its "no" form (`no ip igmp`) disables IGMP on the interface.

Default
IGMP is not run on interfaces by default.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example enables IGMP on interface fxp1 only.

    (config)# interface fxp1
    (config-if)# ip igmp

Example 2
With the command line still in Interface Configuration Mode for interface fxp1, the following example disables IGMP on that interface.

    (config-if)# no ip igmp
    (config-if)# exit
    (config)#
**ip igmp ignore-v1-messages**

**Name**

*ip igmp ignore-v1-messages* - specifies whether to process IGMPv1 messages on the associated interface(s)

**Syntax**

```plaintext
ip igmp ignore-v1-messages
no ip igmp ignore-v1-messages
```

**Mode**

Interface Configuration

**Parameters**

none

**Description**

The `ip igmp ignore-v1-messages` configuration disables processing of all IGMPv1 messages on the associated interface. Note that this breaks interoperability with older IGMPv1 speakers on the network and should be done when it is important to maintain small group leave latencies.

**Note:** This configuration is meaningful only when the current version on the interface is IGMPv2 or IGMPv3. A warning will be given if you attempt to configure `ignore-v1-messages` and the current version is IGMPv1.

The negative form of this command resets the option such that IGMPv1 messages are not ignored.

**Default**

If `ignore-v1-messages` is not configured, it is the same as if the user had specified the following:

```plaintext
(config-if)# no ip igmp ignore-v1-messages
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, `ip igmp ignore-v1-messages` is used to disable processing of IGMPv1 messages on interface fxp0.

```plaintext
(config)# interface fxp0
(config-if)# ip igmp ignore-v1-messages
(config-if)# exit
(config)#
```

**ip igmp ignore-v2-messages**

**Name**

*ip igmp ignore-v2-messages* - specifies whether to process IGMPv2 messages on the associated interface(s)

**Syntax**

```plaintext
ip igmp ignore-v2-messages
no ip igmp ignore-v2-messages
```
**Mode**

Interface Configuration

**Parameters**

none

**Description**

The `ip igmp ignore-v2-messages` configuration disables processing of all IGMPv2 messages on the associated interface. Note that this breaks interoperability with older IGMPv2 speakers on the network and should be done when it is important to maintain small group leave latencies.

**Note:** This configuration is meaningful only when the current version on the interface is IGMPv3. A warning will be given if you attempt to configure `ignore-v2-messages` and the current version is not IGMPv3.

The negative form of this command resets the option such that IGMPv2 messages are not ignored.

**Default**

If `ignore-v2-messages` is not configured, it is the same as if the user had specified the following:

```
(config-if)# no ip igmp ignore-v2-messages
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, `ip igmp ignore-v2-messages` is used to disable processing of IGMPv1 messages on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip igmp ignore-v2-messages
(config-if)# exit
(config)#
```

**ip igmp last-member-query-count**

**Name**

ip igmp last-member-query-count - specifies the number of queries sent out on startup, separated by the Last Member Query Interval

**Syntax**

```
ip igmp last-member-query-count value
no ip igmp last-member-query-count value?
```

**Mode**

Global Configuration

Interface Configuration

**Parameters**

value - an integer between 1 and 65535, inclusive

**Description**

The command configures the value of the Last Member Query Count as specified in RFC 3376. It specifies the number of queries sent out on startup, separated by the Last Member Query Interval.
The **last-member-query-count** and its **no** form configuration can be issued in both Global Configuration and Interface Configuration modes. If this is configured in Global Configuration mode, then it configures the default Last Member Query Count for all interfaces over which IGMP is running. If it is configured in Interface Configuration mode, then it configures the Last Member Query Count for the associated interfaces.

Use the **no** form of the configuration to return to the default **last-member-query-count** value. **Note:** Specifying a value in the **no** form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If an interface-scoped **last-member-query-count** is configured, then that configured value is used on the interface. If an interface-scoped **last-member-query-count** is not configured, then the value used on the interface is the value specified in an interface-scoped **robustness**. If an interface-scoped **robustness** is not configured, then the value used on the interface is the value specified in the globally scoped **last-member-query-count**. If no globally scoped **last-member-query-count** is configured, then the value used on the interface is the value of the globally scoped **robustness**. Finally, if no globally scoped robustness is configured, then the value used on the interface is 2.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures the Last Member Query Count to be 4.

```
(config)# ip igmp last-member-query-count 4
```

**Example 2**

In the following example, the globally scoped Robustness is configured to be 4. The globally scoped Last Member Query count is configured to be 5. Neither the interface-scoped Robustness nor the interface-scoped Last Member Query Count is configured for interface fxp1. The **no** **ip igmp startup-query-count** for fxp1 implies that the globally scoped Last Member Query Count value of 5 will be used on that interface.

```
(config)# ip igmp robustness 4
(config)# ip igmp last-member-query-count 5
(config)# interface fxp1
(config-if)# no ip igmp startup-query-count
(config-if)# exit
(config)#
```

**ip igmp last-member-query-interval**

**Name**

**ip igmp last-member-query-interval** - specifies the maximum amount of time that hosts are allowed to respond to Group-Specific query messages

**Syntax**

```
ip igmp last-member-query-interval time-milliseconds
no ip igmp last-member-query-interval time-milliseconds?
```

**Mode**

Global Configuration
Interface Configuration

**Parameters**

time-milliseconds - specifies a number of milliseconds. This value can be either 0 or an integer from 1000 to 31744000, inclusive.

**Description**

The `ip igmp last-member-query-interval` command configures the length of the Last Member Query Interval - the interval determining the maximum amount of time that hosts are given to respond to Group-Specific or Group-and-Source-Specific queries. This value is encoded in the Max Resp Code field of Group-Specific or Group-and-Source-Specific Query messages and can be either 0 or an integer from 1000 to 31744000, inclusive. **Note:** The value range 1 to 999 is not permitted.

The Last Member Query Interval configuration and its no form can be issued in both Global Configuration and Interface Configuration modes. If this is configured in Global Configuration mode, then it configures the default Last Member Query Interval for all interfaces over which IGMP is running. If this is configured in Interface Configuration mode, then it configures the Last Member Query Interval for the associated interfaces.

Use the no form of this configuration to return to the default value of the Last Member Query Interval. **Note:** Specifying a value for `time-milliseconds` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip igmp last-member-query-interval` is not specified, it is the same as if the user had entered the following:

```
(config)# ip igmp last-member-query-interval 1000
or
(config-if)# ip igmp last-member-query-interval 1000
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures the default Last Member Query Interval to be 2000 milliseconds (20 seconds)

```
(config)# ip igmp last-member-query-interval 2000
```

**Example 2**

The following example configures the default Last Member Query Interval to be 3000 milliseconds. This value is then overridden in interface fxp0, where it is configured to be 2000 milliseconds. The no `ip igmp last-member-query-interval` for interface fxp1 implies that the configured default value of 3000 will be used on that interface.

```
(config)# ip igmp last-member-query-interval 3000
(config)# interface fxp0
(config-if)# ip igmp last-member-query-interval 2000
(config-if)# exit
(config)# interface fxp1
(config-if)# no ip igmp last-member-query-interval
(config-if)# exit
(config)#
```
ip igmp query-interval

Name
ip igmp query-interval - specifies the value of the query interval in seconds

Syntax
ip igmp query-interval time-seconds
no ip igmp query-interval time-seconds?

Mode
Global Configuration
Interface Configuration

Parameters
time-seconds - specifies a number in seconds between 1 and 3174, inclusive. Note: This value cannot be less than the Query Response Interval.

Description
This command specifies the value of the Query Interval in seconds. The Query Interval is the interval between General Queries sent by the Querier. It is encoded in the Query Interval Code (QQIC) field of General Queries. The Query Interval cannot be less than the Query Response Interval (configured using ip igmp query-response-interval). The maximum value of the Query Interval is 3174 seconds.

The Query Interval configuration and its no form configuration can be issued in both Global Configuration Mode and Interface Configuration Mode. If it is configured in Global Configuration Mode, it configures the default Query Interval for all interfaces over which IGMP is running. If it is configured in Interface Configuration Mode (in other words, at the (config-if)# prompt), then it configures the Query Interval for the associated interfaces.

Use the no form of the configuration to return to the default value of 125 seconds. Note: Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If ip igmp query-interval is not specified, it is the same as if the user had specified the following:
(config)# ip igmp query-interval 125
or
(config-if)# ip igmp query-interval 125

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures the default Query Interval for all interfaces to be 200 seconds.
(config)# ip igmp query-interval 200

Example 2
The following example configures the default Query Interval to be 200 seconds and the Query Interval for interface fxp0 to be 210 seconds. The no igmp query-interval command for fxp1 implies that the configured default value of 200 seconds will be used on that interface.
(config)# ip igmp query-interval 200
(config)# interface fxp0
ip igmp query-max-response-time

**Name**

*ip igmp query-max-response-time* - configures the maximum amount of time allowed before a host sends a report message.

**Syntax**

```
ip igmp query-max-response-time time-seconds
no ip igmp query-max-response-time time-seconds?
```

**Mode**

- Global Configuration
- Interface Configuration

**Parameters**

- `time-seconds` - a number between 0 and 3174 seconds, inclusive. Note: This value must be less than or equal to the Query Interval.

**Description**

The `ip igmp query-max-response-time` command configures the length of the Query Response Interval, the maximum amount of time allowed before a Host sends a Report message in response to a received General Query. This interval is encoded in the Max Resp Code field of General Query messages. The interval can have a value from 0 to 3174 seconds. This value must be less than or equal to the Query Interval.

The `ip igmp query-max-response-time` command and its no form can be issued in both Global Configuration Mode and Interface Configuration Mode. When issued in Global Configuration Mode, it configures the default Query Response Interval for all interfaces over which IGMP is running. If it is configured in Interface Configuration Mode, it configures the Query Response Interval for the associated interfaces.

Use the no form of the configuration to return to the default value of 10 seconds. **Note**: Specifying a value for `time-seconds` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Note**: There is forward and backward conversion between the Advanced Routing Suite CLI and the Advanced Routing Suite XML for this item. The equivalent XML command, `<query-response-interval>`, is configured in units of deciseconds. When this value is configured in the Advanced Routing Suite CLI, the CLI converts it from seconds to deciseconds before sending the configuration to XML. Then, in show running, the CLI receives deciseconds from the XML configurator and converts it back to seconds, rounding it down to the nearest second. For example, if this value is configured in XML as 31712 deciseconds, then it will be read in the CLI as 3171 seconds.

**Default**

If `ip igmp query-max-response-time` is not specified, it is the same as if the user had specified the following:

```
(config)# ip igmp query-max-response-time 10
or
(config-if)# ip igmp query-max-response-time 10
```
Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures the default Query Response Interval to be 10 seconds (or 100
deciseconds).

    (config)# ip igmp query-max-response-time 100

Example 2
The following example configures the default Query Response Interval to be 11 seconds and the Query
Response Interval for interface fxp0 to be 12 seconds. The no ip igmp query-max-response-time
command for fxp1 implies that the configured default value of 110 deciseconds will be used on that
interface.

    (config)# ip igmp query-max-response-time 110
    (config)# interface fxp0
    (config-if)# ip igmp query-max-response-time 120
    (config-if)# exit
    (config)# interface fxp1
    (config-if)# no ip igmp query-max-response-time

ip igmp require-router-alert

Name
ip igmp require-router-alert - specifies whether to ignore messages that do not contain the
Router Alert option, thereby improving protocol security

Syntax
    ip igmp require-router-alert
    no ip igmp require-router-router-alert

Mode
Global Configuration
Interface Configuration

Parameters
none

Description
The require-router-alert configuration specifies whether to ignore messages that do not contain the
Router Alert option, thereby improving protocol security. If this command is configured, then the
following messages are ignored if they do not contain the Router Alert option:

- State-Change Report
- Current-State Report
- Leave Message in IGMP version 2 mode
- Report Message in IGMP version 2 mode

This configuration and its no form configuration can be issued in both Global Configuration and Interface
Configuration modes. If this is configured in Global Configuration mode, then it configures the default for all
interfaces over which IGMP is running. If it is configured in Interface Configuration mode, then it configures the parameter for the associated interfaces.

Use the no form configuration to return to the default value.

Default
If require-router-alert is not specified, it is the same as if the user had specified the following:

```
(config)# ip igmp require-router-alert
or
(config-if)# ip igmp require-router-alert
```

Command History
NGC 2.2 - This command was introduced.
NGC 2.3 - The default for this command changed from off to on.

Examples
The following example turns require-router-alert off for interface fxp0 and on for all other interfaces.

```
(config)# interface fxp0
(config-if)# no ip igmp require-router-alert
(config-if)# exit
(config)# ip igmp require-router-alert
```

**ip igmp robustness**

**Name**
ip igmp robustness - allows for tuning of the IGMP protocol or interface to accommodate a lossy subnet

**Syntax**

```
ip igmp robustness value
no ip igmp robustness value?
```

**Mode**
Global Configuration
Interface Configuration

**Parameters**
value - an integer between 2 and 7, inclusive

**Description**
The **ip igmp robustness** command allows you to tune IGMP for the expected loss on a network. IGMP is robust to (Robustness - 1) packet losses, and this value is advertised in the Querier's Robustness Variable (QRV) field of Query.

The **ip igmp robustness** configuration and its no form can be issued in both Global Configuration and Interface Configuration modes. If this command is configured in Global Configuration mode, it configures the default Robustness for all interfaces over which IGMP is running. If it is configured in Interface Configuration mode, it configures the Robustness for the associated interfaces.

Use the no form of this configuration to return to the default value of Robustness. **Note:** Specifying a value in the no form has no effect on the configuration. Thus, it is displayed above as optional.
Default

If ip igmp robustness is not specified, it is the same as if the user had specified the following:

    (config)# ip igmp robustness 2
or
    (config-if)# ip igmp robustness 2

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following example configures the default Robustness to be 3.

    (config)# ip igmp robustness 3

Example 2

The following example configures the default Robustness to be 3 and the Robustness for interface fpx0 to be 4. The no ip igmp robustness for interface fpx1 implies that the configured default value of 3 will be used on that interface.

    (config)# ip igmp robustness 3
    (config)# interface fpx0
    (config-if)# ip igmp robustness 4
    (config-if)# exit
    (config)# interface fpx1
    (config-if)# no ip igmp robustness

ip igmp send-router-alert

Name

ip igmp send-router-alert - specifies whether sent IGMP packets will include the Router Alert option in the IP packet header

Syntax

    ip igmp send-router-alert
    no ip igmp send-router-alert

Mode

Global Configuration
Interface Configuration

Parameters

none

Description

Use the ip igmp send-router-alert option to specify that sent packets should include the router alert option in the IP packet header.
Default

If `ip igmp send-router-alert` is not specified, it is the same as if the user had specified the following:

```
(config)# no ip igmp send-router-alert
```

or

```
(config-if)# no ip igmp send-router-alert
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example turns the send-router-alert option on for all of IGMP, then explicitly turns it off for interface fxp1.

```
(config)# ip igmp send-router-alert
(config)# interface fxp1
(config-if)# no ip igmp send-router-alert
(config-if)# exit
(config)#
```

**ip igmp startup-query-count**

**Name**

`ip igmp startup-query-count` - specifies the number of queries sent out on startup, separated by the Startup Query Interval

**Syntax**

```
ip igmp startup-query-count value
no ip igmp startup-query-count value?
```

**Mode**

Global Configuration

Interface Configuration

**Parameters**

`value` - an integer between 1 and 65535, inclusive

**Description**

This configuration specifies the value of the Startup Query Count as specified in RFC 236 and RFC 3376. It configures the number of queries sent out on startup, separated by the Startup Query Interval. The number of queries can be an integer between 1 and 65535, inclusive.

This configuration, and its `no` form configuration can be issued in both Global Configuration and Interface Configuration modes. If this is configured in Global Configuration mode, then it configures the default Startup Query Count for all interfaces over which IGMP is running. If it is configured in Interface Configuration mode, then it configures the Startup Query Count for the associated interfaces.

Use the `no` form of this configuration to return to the default value of Startup Query Count. **Note:** Specifying a value in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If an interface-scoped startup-query-count is configured, then the configured value is used on the interface. If an interface-scoped startup-query-count is not configured, then the value used on the interface is the value specified in an interface-scoped robustness. If an interface-scoped robustness is
not configured, then the value used on the interface is the value specified in the globally scoped startup-query-count. If no globally scoped startup-query-count is configured, then the value used on the interface is the value of the globally scoped robustness. Finally, if no globally scoped robustness is configured, then the value used on the interface is 2.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures the globally scoped Startup Query Count to be 4.

```config
(ip)# ip igmp startup-query-count 4
```

Example 2
In the following example, the globally scoped Robustness value is configured to be 4. The `no ip igmp startup-query-count` for interface fxp1 implies that the globally scoped Robustness value of 4 will be used on that interface.

```config
(ip)# ip igmp robustness 4
(ip)# interface fxp1
(config-if)# no ip igmp startup-query-count
(config-if)# exit
(ip)#
```

**ip igmp startup-query-interval**

Name

*ip igmp startup-query-interval* - specifies the time between successive General Query messages on startup.

Syntax

```config
ip igmp startup-query-interval time-deciseconds
no ip igmp startup-query-interval time-deciseconds?
```

Mode

Global Configuration

Interface Configuration

Parameters

*time-deciseconds* - a number in tenths of a second between 10 and 317440

Description

This command configures the value of the Startup Query Interval as specified in RFC3376. This parameter determines the time between successive General Query messages on startup. This value is expressed in units of deciseconds (1/100 of a second) and is encoded in the Max Resp Code field of Query messages. The range of this parameter is 10 to 317440 deciseconds. All values between 10 and 1270 deciseconds are exactly configurable. Above 1270 deciseconds, only intervals for which integer values of 'e' and 'm' exist to satisfy the equation below are exactly configurable.

\[
\text{interval} = (16 + m) \times 2^{(e + 3)}
\]

where
0 <= e <= 7
0 <= m <= 15

If no values of m and e exist to exactly represent the configured value, then Advanced Routing Suite will round down to the next lower, representable value is used.

The Startup Query Configuration and its no form configuration can be issued in both Global Configuration and Interface Configuration modes. If this is configured in Global Configuration mode, then it configures the default Startup Query Interval for all interfaces over which IGMP is running. If this is configured in Interface Configuration mode, then it configures the Startup Query Interval for the associated interface.

Use the no form of this configuration to return to the default value of Startup Query Interval. **Note:** Specifying a value for deciseconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If an interface-scoped startup-query-interval is configured, then the configured value is used on the interface. If an interface-scoped startup-query-interval is not configured, then the value used on the interface is one-fourth (1/4) the value specified in an interface-scoped query-interval. If an interface-scoped query-interval is not configured, then the value used on the interface is the value specified in the globally scoped startup-query-interval. If no globally scoped startup-query-interval is configured, then the value used on the interface is the value specified in the globally scoped query-interval. Finally, if no globally scoped query-interval is configured, then the value used on the interface is 31 seconds.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures the default Startup Query Interval to be 120 deciseconds (12 seconds).

```
(config)# ip igmp startup-query-interval 120
```

**Example 2**

The following example, the default Startup Query Interval is configured to be 120 deciseconds. The Startup Query Interval for interface fxp0 is configured to be 110 deciseconds. The no ip igmp startup-query-interval for fxp1 implies that the globally scoped Startup Query Interval value of 120 will be used on that interface.

```
(config)# ip igmp startup-query-interval 120
(config)# interface fxp0
(config-if)# ip igmp startup-query-interval 110
(config-if)# exit
(config)# interface fxp1
(config-if)# no ip igmp startup-query-interval
```

**Example 3**

In the following example, an attempt is made to configure the Startup Query Interval for fxp0 to 274. No values of m and e satisfy the equation below.

\[247 = (16 + m) \times 2^{(e + 3)}\]

Therefore, the value that Advanced Routing Suite will actually use will be the next lowest representable value, 272 \((m = 1, e = 1)\)

Attempted configuration:

```
(config)# interface fxp0
```
(config-if)# ip igmp startup-query-interval 274
(config-if)# exit
(config)#

See Also
"ip igmp query-interval" ("ip igmp query-interval" on page 339)

ip igmp static-group

Name
ip igmp static-group - causes the router to establish a static join to the multicast group

Syntax

   ip igmp static-group group-address [ source source-address ]?
   no ip igmp static-group group-address [ source source-address ]?

Mode
Interface Configuration

Parameters

group group-address - a valid multicast group address in dotted-quad notation

source source-address - a valid multicast source addresses in dotted-quad notation. Specifying this is optional. Note that when specifying a source address, the source keyword is required.

Description
The ip igmp static-group command causes the router to establish a static join to the multicast group. Note that the group is not actually joined. In other words, no IGMP messages are triggered in response to this command. Instead, this command causes multicast routing to behave as if there were a local member of the indicated group (and source) reachable via the associated interface.

The group-address value is the address of the multicast group in dotted-quad notation. The source keyword allows you to specify a (source, group) pair to be forwarded out of the interface if the group is within the SSM range.

Use the negative form of this command, no ip igmp static-group, to cancel a static join for a multicast group.

Default
none

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures the router to join group 224.1.1.1 on interface fxp0.

   (config)# interface fxp0
   (config-if)# ip igmp static-group 224.1.1.1
   (config-if)# exit
   (config)#
Example 2
In the following example, the router is configured to cancel its membership for group 224.1.1.1.

```
(config)# interface fxp0
(config-if)# no ip igmp static-group
(config-if)# exit
(config)#
```

Example 3
In the following example, the router is configured to forward packets from source 192.168.11.1 for group 232.1.1.1 on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip igmp static-group 232.1.1.1 source 192.168.11.1
(config-if)# exit
(config)#
```

**ip igmp trace file**

**Name**
ip igmp trace file - specifies file options when tracing in IGMP

**Syntax**
ip igmp trace file file_name [no-timestamp || overwrite]?
no ip igmp trace file name file_name [no-timestamp || overwrite]?

**Mode**
Global Configuration

**Parameters**
file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
no-timestamp - specifies that a timestamp should not be prepended to all trace lines
overwrite - specifies to begin tracing by appending or truncating an existing file

**Description**
The trace file command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. For IGMP, the ip igmp trace file command specifies a file for tracing of all IGMP events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the ip igmp trace flag command.

The no-timestamp option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The overwrite option specifies whether to start tracing by truncating or appending to an existing file.

**Note:** These options are not cumulative across multiple commands. Consider the following example:

```
(config)# ip igmp trace file /var/log/igmp.log no-timestamp
(config)# ip igmp trace file /var/log/igmp.log
```

The option given in the second command completely replaces that given in the first.
Default
IGMP tracing is turned off by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, IGMP tracing is written to the file "/var/tmp/igmp.log"
(config)# ip igmp trace file /var/tmp/igmp.log

ip igmp trace flag

Name
ip igmp trace flag - specifies IGMP-specific tracing options as well as options that are common across all protocols

Syntax

ip igmp trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ query | report | leave | packets ] [ send | receive | send-receive ]? [detail?] )

no ip igmp trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ query | report | leave | packets ] [ send | receive | send-receive ]? [detail?] )

Mode
Global Configuration

Parameters
Flags common to all protocols:

[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:

- route - trace routing table changes for routes installed by this protocol or peer
- normal - trace normal protocol occurrences. Note: Abnormal protocol occurrences are always traced.
- state - trace state machine transition in the protocol
- policy - trace the application of protocol and user-specified policy to routes being imported or exported
- task - trace system interface and processing associated with this protocol
- timer - trace timer usage by this protocol
- all - turns on all trace flags

IGMP-specific flags:

[ leave | query | report | packets ] - The IGMP-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:

- leave - trace all IGMPv3 Leave messages.
- query - trace IGMPv3 Membership Query messages.
- report - trace IGMPv3 Membership Report messages.
- packets - trace all IGMPv3 packets
  [ send | receive | send-receive ]? - optionally specify whether to limit the tracing to packets sent, received, or both
  [detail?] - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

**Description**

Use the `ip igmp trace flag` command to specify tracing flags for IGMP tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both query and leave packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, trace flags specify that both the sent and received query and report messages are traced in detail. This tracing information will be written to the file `/var/tmp/igmp.log`.

```
(config)# ip igmp trace file /var/tmp/igmp.log
(config)# ip igmp trace flag query send-receive detail
(config)# ip igmp trace flag report send-receive detail
```

### ip igmp version

**Name**

`ip igmp version` - specifies the version of IGMP to run

**Syntax**

```
ip igmp version [ 1 | 2 | 3 ]
no ip igmp version
```

**Mode**

Interface Configuration

**Parameters**

1 | 2 | 3 - specifies whether to run version 1, version 2, or version 3 of IGMP

**Description**

Use the `ip igmp version` configuration to specify the version of IGMP that you want to run.

**Default**

If `ip igmp version` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip igmp version 3
```

**Command History**

NGC 2.2 - This command was introduced.
Examples
The following example configures IGMP version 2 to run on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip igmp version 2
(config-if)# exit
(config)#
```

show ip igmp groups

**Name**
show ip igmp groups - displays IGMP information regarding multicast group membership

**Syntax**
```
show ip igmp groups [group-name || interface
(ipv4_address | interface_name)] [detail]?
show ip igmp groups [group-address || interface
(ipv4_address | interface_name)] [detail]?
```

**Mode**
Privileged Execution

**Parameters**
group-name - the name of the multicast group, as defined in DNS host table. If a group-name is specified, an optional interface can also be specified to view group information for that interface.
group-address - the IPv4 address of the multicast group in dotted-quad format. Entering this is optional. If a group-address is specified, an optional interface can also be specified to view group information for that interface.
interface (ipv4_address | interface_name) - optionally specify to view information for a specific interface or for a specific interface within a specific group
detail - displays IGMPv3 style group information and source information. Entering this is optional. Otherwise, IGMPv2 style group information is given.

**Description**
Use show ip igmp groups to obtain IGMP information regarding multicast group membership. Queries can be submitted in multiple forms, as shown in the examples that follow. If a query is issued without specifying a group, then group membership information for all groups that are connected to this router will be returned. Alternatively, a query can be issued with the name of the multicast group, the IPv4 address of the multicast group, and/or the interface specified, with or without the detail option. If detail is specified, the response will contain IGMPv3 specific information, including the group mode and other source information.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example queries information pertaining to IGMP or all groups on all interfaces.

```
# show ip igmp groups
```
Example 2

The following example is a query for group information pertaining to IGMP for multicast group 224.1.1.1.

```
# show ip igmp groups 224.1.1.1
```

---

Example 3

The following query requests group information pertaining to IGMP for interface eth1.

```
# show ip igmp groups interface eth1
```

---

Example 4

The following query is a request for group information pertaining to group 224.1.1.1 on interface eth1.

```
# show ip igmp groups 224.1.1.1 interface eth1
```

---

Example 5

The following example is a query for group information pertaining to IGMP in detail for all groups on all interfaces.

```
# show ip igmp groups detail
```

---
Interface: Ethernet0/2 (192.168.13.1)
Group: 224.3.3.3
Uptime: 00:00:34
Expires: 00:00:40
Last reporter: 192.168.13.2
Last-member-query-counter: 0
Last-member-query-timer-expiry: off
Group mode: INCLUDE
Version1-host-present-timer-expiry: 00:00:20
Version2-host-present-timer-expiry: off
Source list:
  Source: 10.1.1.1
  Uptime: 00:00:34
  Expires: 00:00:40
  Last reporter: 192.168.13.2
  Last-member-query-counter: 0
  Last-member-query-timer-expiry: off
  Source: 10.3.3.3
  Uptime: 00:00:34
  Expires: 00:00:40
  Last reporter: 192.168.13.2
  Last-member-query-counter: 1
  Last-member-query-timer-expiry: 00:00:02

Example 6
In the following example, group information pertaining to group 224.1.1.1 on interface eth3, in detail, is requested.

  # show ip igmp groups 224.1.1.1 interface eth3 detail
  IGMP connected Group Membership
  Interface: Ethernet0?1 (192.168.13.2)
  Group: 224.1.1.1
  Uptime: 00:00:34
  Expires: 00:00:40
  Last reporter: 192.168.13.2
  Last-member-query-counter: 0
  Last-member-query-timer-expiry: off
  Source list
    Source: 10.1.1.1
    Uptime: 00:00:34
    Expires: 00:00:40
    Last reporter: 192.168.13.2
    Last-member-query-counter: 0
    Last-member-query-timer-expiry: off
    Source: 10.3.3.3
    Uptime: 00:00:34
    Expires: 00:00:40
    Last reporter: 192.168.13.2
    Last-member-query-counter: 1
    Last-member-query-timer-expiry: 00:00:02

Example 7
In the following example, group information in details pertaining to group 224.1.1.1 on interface eth3 is requested. The response indicates that IGMP is not running on the router.

  # show ip igmp groups 224.1.1.1 interface eth3 detail
  IGMP is not running.
Field Descriptions
The following table describes the fields that appear in the IGMP Group Information Query.

Table 21-1   IGMP Group Information Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>The IPv4 address of the multicast group</td>
</tr>
<tr>
<td>Interface</td>
<td>The interface through which the group has been joined</td>
</tr>
<tr>
<td>Uptime</td>
<td>The length of time, in hours, minutes, and seconds, that this multicast group has been learned</td>
</tr>
<tr>
<td>Expires</td>
<td>The length of time, in hours, minutes, and seconds, until the entry is to be removed from the group table</td>
</tr>
<tr>
<td>Last Reporter</td>
<td>The last host that reported being a member of the multicast group</td>
</tr>
<tr>
<td>Last-member-query-counter</td>
<td>The remaining number of group-specific or group-and-source specific queries to be sent over the interface for the group or source</td>
</tr>
<tr>
<td>Last-member-query-timer-expiry</td>
<td>The length of time in hours, minutes, and seconds until the next group-specific or group-and-source specific query is to be sent</td>
</tr>
<tr>
<td>Group mode</td>
<td>The router filter mode for the group</td>
</tr>
<tr>
<td>Version1-host-present-timer-expiry</td>
<td>The length of time in hours, minutes, and seconds until the Version 1 Host Present Timer will expire</td>
</tr>
<tr>
<td>Version2-host-present-timer-expiry</td>
<td>The length of time in hours, minutes, and seconds until the Version 2 Host Present Timer will expire</td>
</tr>
<tr>
<td>Source</td>
<td>The IPv4 address of the source</td>
</tr>
</tbody>
</table>
show ip igmp interface

Name
show ip igmp interface - displays multicast-related information about a specific interface or all interfaces

Syntax
    show ip igmp interface [ interface-name | ipv4_address ]?

Mode
User Execution

Parameters
interface-name - optionally specify a physical interface
ipv4_address - optionally specify the logical interface address

Description
The show ip igmp interface query displays multicast-related information about a specific interface or all interfaces.

This query has two forms. If the query is issued without arguments, then information on about all interfaces over which IGMP is running is returned. Alternatively, the query can be issued with a specific interface. If this is the case, then the reply will contain information pertaining only to that referenced interface.

Note that if interfaces are added to the set of interfaces over which IGMP is running after this query has been issued but before the query is finished, it is not guaranteed that the new interfaces will be reported. Similarly, if IGMP is de-configured on an interface after the query has been issued but before the query is finished, then the interface may or may not be reported.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example displays interface information for all interfaces on which IGMP is running.

    # show ip igmp interface
    eth0 is up
    Internet address: 10.129.10.26/9
    IGMP on this interface: disabled
    Multicast routing on this interface: disabled
    Multicast TTL threshold: 0
    Multicast groups joined: 0

    eth1 is up
    Internet address: 192.168.11.1/24
    IGMP on this interface: enabled
    Current IGMP router version: 2
    IGMP query interval: 60 seconds
    IGMP max query response time: 10 seconds
    Last member query response interval: 1000 ms
    IGMP activity: 1 joins, 0 leaves
    Multicast routing on this interface: enabled
    Multicast TTL threshold: 0
    Multicast designated router (DR): 192.168.11.1 (this system)
    IGMP Querier: 192.168.11.1 (this system)
Robustness: 2
Require-router-alert: no
Last-member-query-interval: 30 deciseconds
Startup-query-interval: 300 deciseconds
Startup-query-count: 2
General-query-timer-expiry: 00:00:20 (hours:minutes: seconds)
Startup-query-timer-expiry: off
Other-querier-present-timer-expiry: off
Multicast groups joined (number of users):
  224.0.0.2(1)
  224.1.1.1(1)

fxp0 is up
Internet address: 192.168.12.1/24
IGMP on this interface: enabled
Current IGMP router version: 3
IGMP query interval: 125 seconds
IGMP max query response time: 10 seconds
Last member query response interval: 1000 ms
IGMP activity: 1 joins, 0 leaves
Multicast routing on this interface: enabled
Multicast TTL threshold: 0
Multicast designated router (DR): 1922.168.12.1 (this system)
IGMP Querier: 129.168.12.1 (this system)
Robustness: 2
Require-router-alert: no
Last-member-query-interval: 30 deciseconds
Startup-query-interval: 300 deciseconds
Startup-query-count: 2
General-query-timer-expiry: 00:00:20 (hours:minutes: seconds)
Startup-query-timer-expiry: off
Other-querier-present-timer-expiry: off
Multicast groups joined (number of users):
  224.0.0.2(1)
  224.2.2.2(1)

Example 2
In the following example, a query is issued for interface eth1 only. The reply indicates that IGMP is running on this interface, and it has two groups joined.

    # show ip igmp interface eth1
    eth1 is up
    Internet address: 192.168.11.1/24
    IGMP on this interface: enabled
    Current IGMP router version: 2
    IGMP query interval: 60 seconds
    IGMP max query response time: 10 seconds
    Last member query response interval 1000 ms
    IGMP activity: 1 joins, 0 leaves
    Multicast routing on this interface: enabled
    Multicast TTL threshold: 0
    Multicast designated router (DR): 192.168.11.1 (this system)
    IGMP Querier: 192.168.11.1
    Robustness: 2
    Require-router-alert: no
    Last-member-query-interval: 30 deciseconds
    Startup-query-interval: 300 deciseconds
    Startup-query-count: 2
    General-query-timer-expiry: 00:00:20 (hours:minutes: seconds)
Example 3
In the following example, a query is issued for interface eth2 only. The reply indicates that IGMP is not running on the router.

```
# show ip igmp interface eth2
IGMP is not running
```

Field Descriptions
The following table describes the fields that appear in the IGMP Interface Information Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0/1 is up, line protocol is up</td>
<td>Interface type, number, and status</td>
</tr>
<tr>
<td>Internet address</td>
<td>The IPv4 address of the interface</td>
</tr>
<tr>
<td>IGMP is enabled on interface</td>
<td>Shows whether IGMP is enabled on an interface</td>
</tr>
<tr>
<td>Current IGMP router version</td>
<td>IGMP router version</td>
</tr>
<tr>
<td>IGMP query interval</td>
<td>Query interval in seconds</td>
</tr>
<tr>
<td>IGMP robustness</td>
<td>Robustness value</td>
</tr>
<tr>
<td>IGMP max query response interval</td>
<td>Query Response Interval in seconds</td>
</tr>
<tr>
<td>Last member query response interval</td>
<td>Last Member Query Response Interval in milliseconds</td>
</tr>
<tr>
<td>IGMP activity: joins, leaves</td>
<td>Activity statistics showing the number of joins and leaves</td>
</tr>
<tr>
<td>Multicast routing on this interface</td>
<td>Indicates whether multicast routing is enabled</td>
</tr>
<tr>
<td>Multicast TTL threshold</td>
<td>The packet time-to-live threshold</td>
</tr>
<tr>
<td>Multicast designated router (DR)</td>
<td>IPv4 address of the Designated Router</td>
</tr>
<tr>
<td>IGMP Querier</td>
<td>IPv4 address of the Querier</td>
</tr>
<tr>
<td>Robustness</td>
<td>Robustness value</td>
</tr>
<tr>
<td>Require-router-alert</td>
<td>Indicates whether the require-router-alert option is on</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Last-member-query-interval</td>
<td>Last Member Query interval in deciseconds</td>
</tr>
<tr>
<td>Startup-query-interval</td>
<td>Startup Query Interval in deciseconds</td>
</tr>
<tr>
<td>Startup-query-count</td>
<td>Number of General Queries to send during startup</td>
</tr>
<tr>
<td>General-query-timer-expiry</td>
<td>The length of time, in hours, minutes, and seconds, until another General Query is sent</td>
</tr>
<tr>
<td>Startup-query-timer-expiry</td>
<td>The length of time, in hours, minutes, and seconds, until another General Query is sent during startup</td>
</tr>
<tr>
<td>Other-querier-present-timer-expiry</td>
<td>The length of time, in hours, minutes, and seconds, until the Other Querier Present Timer expires</td>
</tr>
<tr>
<td>Multicast groups joined (number of users)</td>
<td>Group address and number of hosts that joined the group</td>
</tr>
</tbody>
</table>

**show ip igmp interface-summary**

**Name**

show ip igmp interface-summary - **displays simple summary information pertaining to IGMP interfaces**

**Syntax**

```
show ip igmp interface-summary
```

**Mode**

User Execution

**Parameters**

none

**Description**

The **show ip igmp interface-summary** command displays the number of interfaces on which IGMP is running.

**Command History**

NGC 2.2 - This command was introduced.
Examples

Example 1
The following example issues an interface summary query. The response indicates that IGMP is running on five interfaces.

```bash
# show ip igmp interface-summary
Number of active interfaces: 5
```

Example 2
The following example issues an interface summary query on a router on which IGMP is not running.

```bash
# show ip igmp interface-summary
IGMP is not running
```

show ip igmp static-groups

Name
show ip igmp static-groups - displays IGMP information regarding multicast static-group state

Syntax
```bash
show ip igmp static-groups
```

Mode
User Execution

Parameters
none

Description
The `show ip igmp static-groups` query displays information about all configured IGMP multicast static groups.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example displays information pertaining to IGMP for all groups on all interfaces.

```bash
> show ip igmp static-groups
IGMP CONNECTED Static-Group Membership
<table>
<thead>
<tr>
<th>Group Address</th>
<th>Interface</th>
<th>Expires</th>
<th>Last Reporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>224.1.1.1</td>
<td>fxp1</td>
<td>never</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>224.1.1.1</td>
<td>fxp0</td>
<td>never</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>224.1.1.2</td>
<td>fxp0</td>
<td>never</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>224.1.1.3</td>
<td>fxp1</td>
<td>never</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>
```
Chapter 17

Protocol Independent Multicast (PIM)

In This Chapter

- Protocol Independent Multicast (PIM) Overview 360
- ip pim assert-holdtime 361
- ip pim dr-priority 362
- ip pim hello-holdtime 363
- ip pim hello-interval 363
- ip pim jx-holdtime 364
- ip pim jx-interval 365
- ip pim lan-delay 366
- ip pim mrt-interval 367
- ip pim mrt-stale-multiplier 367
- ip pim override-interval 368
- ip pim triggered-hello-delay 369
- show ip pim control-counters 370
- show ip pim interface 371
- show ip pim neighbor 373
- Protocol Independent Multicast - Dense Mode (PIM-DM) 375
- Protocol Independent Multicast - Sparse Mode (PIM-SM) 389

Protocol Independent Multicast (PIM) Overview

Traditional multicast routing mechanisms (for example, DVMRP and MOSPF) were intended for use within regions where groups are densely populated or bandwidth is universally plentiful. When groups, and senders to these groups, are distributed sparsely across a wide area, these "dense mode" schemes do not perform efficiently. PIM is made of two protocols, one for each type of group distribution. PIM Sparse Mode, PIM-SM, provides efficient routing for a group distributed sparsely across a wide area. PIM Dense Mode, PIM-DM, provides multicast routing for a densely populated group.

Multicasting protocols require two different functions in order to create source-based trees or group-based trees:

- a set of routes used to calculate the reverse path forwarding
- a mechanism by which to build trees

PIM is protocol independent because it depends on existing unicast routes to calculate the reverse path forwarding. In contrast, DVMRP passes this set of routes within the protocol.

There are two versions of the PIM-SM protocol. PIM-SM version 1 is documented in RFC 2117. PIM-SM version 2 was constructed to address some of the shortcomings of PIM-SM version 1. Advanced Routing Suite implements only version 2, which is an RFC but is not considered complete enough to implement (RFC 2362). In going from draft-ietf-pim-sm-v2-new-01 to draft-ietf-pim-sm-v2-new-02, the BSR functionality was removed and placed in its own internet draft. Advanced Routing Suite implements the PIM-SM protocol as described in draft-ietf-pim-sm-v2-new-02, but the BSR functionality as described in draft-ietf-pim-sm-v2-new-01.

Note: PIM uses routes in the multicast Routing Information Base (RIB) to perform its RPF check. By default none of the IGPs (IS-IS, OSPF, RIP) place routes into the multicast RIB. They must be configured to do so. Check the relevant protocol's configuration section in order to obtain the correct syntax.
**ip pim assert-holdtime**

**Name**

`ip pim assert-holdtime` - specifies the number of seconds that Assert state should be maintained in the absence of a refreshing Assert message

**Syntax**

```
ip pim assert-holdtime time-seconds
no ip pim assert-holdtime time-seconds?
```

**Mode**

Global Configuration
Interface Configuration

**Parameters**

`time-seconds` - specifies a time, in seconds, between 1 and 1,073,741,823, inclusive

**Description**

When a PIM router receives an Assert message, it modifies the outgoing interface list for a (*,G) or (S,G) entry, as specified by the message. The lifetime of this modification is specified by the `ip pim assert-holdtime` command. If another Assert message does not refresh the Assert state before the lifetime expires, then the outgoing interface list reverts to its previous state.

The negative of this command, `no ip pim assert-holdtime`, removes the configured value and returns this command to its default value. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim assert-holdtime` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip pim assert-holdtime 180
or
(config)# ip pim assert-holdtime 180
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures a global assert-holdtime value of 100. For interface eth1, this value is overridden to be 140.

```
(config)# ip pim assert-holdtime 100
(config)# interface eth1
(config-if)# ip pim assert-holdtime 140
```

**Example 2**

The following example uses the negative form of the assert-holdtime command both globally and for interface eth1 to return the value to 180 seconds.

```
(config)# no ip pim assert-holdtime
(config)# interface eth1
```
(config-if)# no ip pim assert-holdtime

**ip pim dr-priority**

**Name**

*ip pim dr-priority* - sets the priority for determining the designated router (DR)

**Syntax**

```
ip pim dr-priority level
no ip pim dr-priority level?
```

**Mode**

Global Configuration

Interface Configuration

**Parameters**

`level` - an integer between 1 and 4,294,967,295, inclusive

**Description**

PIM Hello messages can contain a priority field that is used to elect a designated router (DR) on a shared network. All Hello messages originated by Advanced Routing Suite contain such a priority. DRs are responsible for encapsulating multicast data from local sources into PIM-SM register messages and for unicasting them to the Rendezvous Point. The router with the highest priority wins the DR election. In the case of a tie, the router with the highest IP address wins.

If at least one neighbor on the network does not use Hello priorities, then election of a DR is carried out using only IP addresses, where the highest address wins.

The negative form of this command, `no ip pim dr-priority`, removes the configured priority and returns it to its default value of 1. **Note:** Specifying a value for `level` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim dr-priority` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip pim dr-priority 1
or
(config)# ip pim dr-priority 1
```

**Command History**

NGC 2.2 - This command was introduced.

NGC 2.3 - The lower limit of this command was changed from 1 to 0.

**Examples**

The following example configures a default dr-priority of 4. This value is then overridden on interface 192.168.22.1 to be 3.

```
(config)# ip pim dr-priority 4
(config)# interface 192.168.22.1
(config-if)# ip pim dr-priority 3
```
**ip pim hello-holdtime**

**Name**

*ip pim hello-holdtime* - specifies how long neighbors should wait for Hello messages before expiring the sender’s neighbor state.

**Syntax**

```
ip pim hello-holdtime time-seconds
no ip pim hello-holdtime time-seconds?
```

**Mode**

Global Configuration

**Interface Configuration**

**Parameters**

*time-seconds* - specifies a time in seconds between 1 and 65535, inclusive

**Description**

PIM Hello messages contain a holdtime specifying how long neighbors must wait for Hello messages before expiring the sender’s neighbor state. The *ip pim hello-holdtime* command specifies the holdtime, in seconds, to advertise in Hello messages.

The negative of this command, *no ip pim hello-holdtime*, removes the configured setting and returns this value to the default of 105 seconds. **Note:** Specifying a value for *time-seconds* in the *no* form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If *ip pim hello-holdtime* is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim hello-holdtime 105
or
(config-if)# ip pim hello-holdtime 105
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the global hello holdtime to be 100 seconds and the holdtime for interface eth1 to be 140 seconds.

```
(config)# ip pim hello-holdtime 100
(config)# interface eth1
(config-if)# ip pim hello-holdtime 140
```

**ip pim hello-interval**

**Name**

*ip pim hello-interval* - specifies the frequency with which Hello messages are sent.

**Syntax**

```
ip pim hello-interval time-seconds
no ip pim hello-interval time-seconds?
```
Mode
Global Configuration
Interface Configuration

Parameters

time-seconds - specifies a time, in seconds, between 1 and 65535, inclusive

Description

PIM routers periodically multicast Hello messages on each network to which they are connected to alert
other routers to the presence of the sender. The `ip pim hello-interval` command specifies the time,
in seconds, between successive Hello messages.

The negative of this command, `no ip pim hello-interval`, removes the configured setting and
returns this value to the default of 30 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has
no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip pim hello-interval` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim hello-interval 30
```

or

```
(config-if)# ip pim hello-interval 30
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the global hello-interval to be 100 seconds. This value is overridden to be
140 seconds on interface eth1.

```
(config)# ip pim hello-interval 100
(config)# interface eth1
(config-if)# ip pim hello-interval 140
```

**ip pim jp-holdtime**

Name

`ip pim jp-holdtime` - specifies the holdtime that is advertised on PIM Join/Prune messages

Syntax

```
ip pim jp-holdtime time-seconds
no ip pim jp-holdtime time-seconds?
```

Mode

Global Configuration
Interface Configuration

Parameters

`time-seconds` - specifies a time in seconds between 1 and 65535, inclusive
Description
The `ip pim jp-holdtime` command specifies the holdtime that is advertised in PIM Join/Prune messages. Receivers must wait at least this long after receiving a Join/Prune message before deleting the Join/Prune state associated with the advertiser. The recommend value is 3.5 * ip-interval.

The negative form of this command, `no ip pim jp-holdtime`, removes the configured value and returns this to its default value of 210 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `ip pim jp-holdtime` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim jp-holdtime 210
or
(config-if)# ip pim jp-holdtime 210
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the default `jp-holdtime` to be 100 seconds and the `jp-holdtime` on interface `eth1` to be 140 seconds.

```
(config)# ip pim jp-holdtime 100
(config)# interface eth1
(config-if)# ip pim jp-holdtime 140
```

**ip pim jp-interval**

Name
`ip pim jp-interval` - specifies the number of seconds between successive Join/Prune messages sent to upstream neighbors

Syntax
```
ip pim jp-interval time-seconds
no ip pim jp-interval time-seconds?
```

Mode
Global Configuration
Interface Configuration

Parameters
`time-seconds` - specifies a time in seconds between 1 and 65535, inclusive

Description
The Join/Prune state on an upstream neighbor must be refreshed by periodic Join/Prune messages. The `ip pim jp-interval` command specifies the number of seconds between successive Join/Prune messages sent to upstream neighbors to maintain the neighbor’s Join/Prune state.

The negative form of this command, `no ip pim jp-interval`, removes the configured value and returns the value to its default. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
Default
If ip pim jp-interval is not specified, it is the same as if the user had specified the following:

(config)# ip pim jp-interval 60
or
(config-if)# ip pim jp-interval 60

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the default jp-interval to be 100 seconds and the jp-interval on interface eth1 to be 140 seconds.

(config)# ip pim jp-interval 100
(config)# interface eth1
(config-if)# ip pim jp-interval 100

ip pim lan-delay

Name
ip pim lan-delay - specifies the value to advertise as the LAN Delay value in the Lan Prune Delay option in PIM Hello messages

Syntax
ip pim lan-delay time-milliseconds
no ip pim lan-delay time-milliseconds?

Mode
Interface Configuration

Parameters
time-milliseconds - an integer between 1 and 65535, specifying a number of seconds

Description
Use the ip pim lan-delay command to configure the value to advertise as the LAN Delay value in the Lan Prune Delay option in PIM Hello messages. This value is used to tune the value of the J/P Override interval. It is configured in units of milliseconds.

The negative form of this command, no ip pim lan-delay, removes the configured time-milliseconds value and returns this to its default value of 500 milliseconds. Note: Specifying a value for time-milliseconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If ip pim lan-delay is not specified, it is the same as if the user had specified the following:

(config-if)# ip pim lan-delay 500

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the Lan Delay for interface eth1 to be 700.
(config)# interface eth1
(config-if)# ip pim lan-delay 700
(config-if)# exit

**ip pim mrt-interval**

**Name**

`ip pim mrt-interval` - specifies the number of seconds to wait between examinations of a PIM component's multicast routing table (MRT)

**Syntax**

```
ip pim mrt-interval time-seconds
no ip pim mrt-interval time-seconds?
```

**Mode**

Global Configuration

**Parameters**

`time-seconds` - an integer between 1 and 3600, inclusive, specifying a number of seconds

**Description**

A PIM component's MRT is examined periodically in order to remove entries that have been marked for deletion. The `ip pim mrt-interval` command specifies the number of seconds to wait between examinations. This can be a computationally expensive operation if the number of entries is large.

The negative form of this command, `no ip mrt-interval`, removes the configured `time-seconds` value and returns this to its default value of 15 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim mrt-interval` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim mrt-interval 15
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the MRT interval to be 100 seconds.

```
(config)# ip pim mrt-interval 100
(config)#
```

**ip pim mrt-stale-multiplier**

**Name**

`ip pim mrt-stale-multiplier` - together with the `ip pim mrt-interval` command, specifies the minimum number of seconds that a source can be silent before its corresponding (S,G) entry can be timed out.
Syntax

ip pim mrt-stale-multiplier multiplier
no ip pim mrt-stale-multiplier multiplier?

Mode
Global Configuration

Parameters

multiplier – an unsigned integer between 1 and 100, inclusive

Description

When a source stops sending to a group, the corresponding (S,G) entry is said to have become "stale" and becomes a candidate for deletion from the PIM multicast forwarding table (MRT). The PIM-SM protocol defines a Keep-Alive Timer (KAT) for each (S,G) entry, which is reset by the arrival of data from source S addressed to group G. If the KAT ever expires, and if other conditions are met, then the entry can be deleted. The ip pim mrt-stale-multiplier command, together with the ip pim mrt-interval command, specifies the minimum number of seconds that a source must be silent before the entry is considered stale. The value, multiplier, specified in the ip pim mrt-stale-multiplier command indicates that a source must be silent for \( m \) * ip pim mrt-interval in order to be declared stale.

The negative form of this command, no ip pim mrt-stale-multiplier, removes the configured multiplier value and returns this to its default value of 14 intervals. **Note:** Specifying a value for multiplier in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Defaults

If ip pim mrt-stale-multiplier is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim mrt-stale-multiplier 14
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the MRT stale multiplier value to be 10 seconds.

```
(config)# ip pim mrt-stale-multiplier 10
(config)#
```

**ip pim override-interval**

**Name**

ip pim override-interval – specifies the value to advertise as the Override Interval in the Lan Prune Delay option of PIM Hello messages

**Syntax**

ip pim override-interval time-milliseconds
no ip pim override-interval time-milliseconds?

**Mode**

Interface Configuration
Parameters

time-milliseconds – an integer between 1 and 65535, specifying a number of seconds

Description

Use the `ip pim override-interval` command to configure the value to advertise as the Override Interval value in the Lan Prune Delay option in PIM Hello messages. This value is used to tune the value of the J/P Override interval. It is configured in units of milliseconds.

The negative form of this command, `no ip pim override-interval`, removes the configured `time-milliseconds` value and returns this to its default value of 2500 milliseconds. **Note:** Specifying a value for `time-milliseconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip pim override-interval` is not specified, it is the same as if the user had specified the following:

```
(config-if)# ip pim override-interval 2500
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the Lan Delay for interface eth1 to be 2900 milliseconds.

```
(config)# interface eth1
(config-if)# ip pim override-interval 2900
(config-if)# exit
```

`ip pim triggered-hello-delay`

Name

`ip pim triggered-hello-delay` – specifies the number of seconds to delay between triggered Hello messages

Syntax

```
ip pim triggered-hello-delay time-seconds
no ip pim triggered-hello-delay time-seconds?
```

Mode

Interface Configuration

Parameters

time-seconds – an integer from 1 to 65535, inclusive

Description

The `ip pim triggered-hello-delay` command configures the maximum delay for a triggered Hello message. The negative form of this command, `no ip pim triggered-hello-delay`, removes the configured `time-seconds` value and returns this to its default value of 5 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip pim triggered-hello-delay` is not specified, it is the same as if the user had specified the following:
(config-if)# ip pim triggered-hello-delay 5

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the Triggered Hello delay to be 10 seconds on interface fxp3.

  (config)# interface fxp3
  (config-if)# ip pim triggered-hello-delay 10
  (config-if)# exit
  (config)#

show ip pim control-counters

Name
show ip pim control-counters – displays counts for both sent and received PIM control packets

Syntax
  show ip pim control-counters

Mode
User Execution

Parameters
  none

Description
The show ip pim control-counters query displays counts for both sent and received PIM control packets on all interfaces.

Command History
NGC 2.2 - This command was introduced.

Examples
The following is a request to return information about all PIM packets.

  > show ip pim control-counters

  PIM Control Counters
  Received      Sent          Invalid
  Assert         10            5             0
  Graft          20            37            2
  Graft Ack      25            20            1
  Hello          1232          453           0
  Joins          5             10            20
  Prunes         10            20            1
  State Refresh  8             7             1

Field Descriptions
The following table describes the fields that appear in the PIM Control Packet Counters Query.

Table 22-1  PIM Control Packet Counters Query Fields
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received</td>
<td>Number of packets of the given type that have been received</td>
</tr>
<tr>
<td>Sent</td>
<td>Number of packets of the given type that have been sent</td>
</tr>
<tr>
<td>Invalid</td>
<td>Number of packets of the given type that were received but were invalid</td>
</tr>
</tbody>
</table>

### show ip pim interface

**Name**

`show ip pim interface` - displays information about the interfaces on which PIM is running

**Syntax**

```
show ip pim interface [ipv4_addr | name]? detail?
```

**Mode**

User Execution

**Parameters**

- `ipv4_addr | name` - optionally specify either an IPv4 address or a physical interface name
- `detail` - displays IGMPv3 style group information and source information. Specifying this is optional.

**Description**

Use `show ip pim interface` to obtain information about a specific interface or all interfaces on which PIM is running.

This query has two forms. If the query is issued without arguments, then information on about all interfaces over which PIM is running is returned. Alternatively, the query can be issued with a specific interface. If this is the case, then the reply will contain information pertaining only to that referenced interface.

Note that if interfaces are added to the set of interfaces over which PIM is running after this query has been issued but before the query is finished, it is not guaranteed that the new interfaces will be reported. Similarly, if PIM is de-configured on an interface after the query has been issued but before the query is finished, then the interface may or may not be reported.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example displays interface information for all interfaces on which PIM is running.

```
> show ip pim interface
Address   Interface Mode Neighbor Hello    DR
Count    Interval
192.168.10.1 fxp0 Dense 5       30   192.168.10.1
10.1.57.28 ex0    Sparse 2      50   10.1.57.30
```
Example 2
The following example displays detailed interface information for interface fxp0.

```bash
> show ip pim interface fxp0 detail
fxp0 is up, line protocol is up
Internet address is 192.168.10.1/25
Multicast packets in/out: 0/0
Multicast boundary: not set
Multicast TTL threshold: 1
PIM: enabled
  PIM version: 2, mode: dense
  PIM DR: 192.168.10.1 (this system)
  PIM Neighbor Count: 5
  PIM Hello/Query Interval: 30 seconds
  PIM State-Refresh processing: enabled
  PIM State-Refresh origination: enabled,
    interval 60 seconds
  PIM domain border: disabled
```

Field Descriptions
The following table describes the fields that appear in the PIM Interface Information Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Shows the IP address used by PIM on the interface</td>
</tr>
<tr>
<td>Interface</td>
<td>The physical interface name</td>
</tr>
<tr>
<td>Mode</td>
<td>The PIM mode active on the interface</td>
</tr>
<tr>
<td>Neighbor Count</td>
<td>Number of PIM neighbors reached through the interface</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>The number of seconds at which hello packets are sent</td>
</tr>
<tr>
<td>DR</td>
<td>The IP address of the elected designated router on the interface</td>
</tr>
<tr>
<td>&lt;interface-name&gt;</td>
<td>The physical name of the interface and the administrative state of it (up or down)</td>
</tr>
<tr>
<td>line protocol</td>
<td>Carrier status of interface (up or down)</td>
</tr>
<tr>
<td>Internet address</td>
<td>The IPv4 address used by PIM on the interface</td>
</tr>
<tr>
<td>Multicast packets</td>
<td>The number of multicast packets received from/forwarded out of this interface</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Multicast boundary</td>
<td>Indicates whether any administratively scoped boundaries are set on the interface</td>
</tr>
<tr>
<td>Multicast TTL</td>
<td>The time-to-live threshold used on the interface</td>
</tr>
<tr>
<td>PIM</td>
<td>Indicates whether PIM is enabled or disabled</td>
</tr>
<tr>
<td>PIM version</td>
<td>Indicates the PIM version configured on the interface</td>
</tr>
<tr>
<td>mode</td>
<td>Indicates whether PIM dense or sparse mode is configured on the interface</td>
</tr>
<tr>
<td>PIM DR</td>
<td>The IPv4 address of the elected designated router on the interface</td>
</tr>
<tr>
<td>PIM Neighbor Count</td>
<td>The number of PIM neighbors reached through this interface</td>
</tr>
<tr>
<td>PIM Hello/Query Interval</td>
<td>The PIM hello interval configured on the interface</td>
</tr>
<tr>
<td>PIM State-Refresh processing</td>
<td>Indicates whether the processing of PIM state refresh messages is enabled or disabled on the interface</td>
</tr>
<tr>
<td>PIM State-Refresh origination</td>
<td>Indicates whether the origination of PIM state refresh messages is enabled or disabled on the interface</td>
</tr>
<tr>
<td>interval</td>
<td>The number of seconds between PIM state refresh messages</td>
</tr>
<tr>
<td>PIM domain border</td>
<td>Indicates whether the interface is a PIM domain border</td>
</tr>
</tbody>
</table>

**show ip pim neighbor**

**Name**

*show ip pim neighbor* - displays information about a router’s PIM neighbors

**Syntax**

`show ip pim neighbor name?`

**Mode**

User Execution
### Parameters

- `name` - optionally specify an interface name

### Description

The `show ip pim neighbor` query displays information about a router's PIM neighbors on all interfaces or on a specific interface.

### Command History

NGC 2.2 - This command was introduced.

### Examples

#### Example 1

The following example displays information about all of this router's PIM neighbors.

```plaintext
> show ip pim neighbor

<table>
<thead>
<tr>
<th>Neighbor Address</th>
<th>Interface</th>
<th>Uptime</th>
<th>Expires</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.1</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
<tr>
<td>192.168.10.2</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense (DR)</td>
</tr>
<tr>
<td>192.168.10.3</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
<tr>
<td>192.168.10.4</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
<tr>
<td>10.0.0.10</td>
<td>ex0</td>
<td>00:00:20</td>
<td>00:01:25</td>
<td>Sparse</td>
</tr>
<tr>
<td>10.0.0.20</td>
<td>ex0</td>
<td>00:00:20</td>
<td>00:01:25</td>
<td>Sparse</td>
</tr>
<tr>
<td>10.0.0.21</td>
<td>ex0</td>
<td>00:00:20</td>
<td>00:01:25</td>
<td>Sparse (DR)</td>
</tr>
<tr>
<td>10.0.0.22</td>
<td>ex0</td>
<td>00:00:20</td>
<td>00:01:25</td>
<td>Sparse</td>
</tr>
</tbody>
</table>
```

#### Example 2

The following example displays information about this router's PIM neighbors on interface fxp0.

```plaintext
> show ip pim neighbor fxp0

<table>
<thead>
<tr>
<th>Neighbor Address</th>
<th>Interface</th>
<th>Uptime</th>
<th>Expires</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.1</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
<tr>
<td>192.168.10.2</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense (DR)</td>
</tr>
<tr>
<td>192.168.10.3</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
<tr>
<td>192.168.10.4</td>
<td>fxp0</td>
<td>10:23:43</td>
<td>00:01:10</td>
<td>Dense</td>
</tr>
</tbody>
</table>
```
Chapter 18

Protocol Independent Multicast - Dense Mode (PIM-DM)

In This Chapter

- Protocol Independent Multicast - Dense Mode (PIM-DM) Overview
- ip pim dense-mode
- ip pim graft-retry-interval
- ip pim require-genid
- ip pim source-lifetime
- ip pim state-refresh-capable
- ip pim state-refresh-interval
- ip pim state-refresh-rate-limit
- ip pim state-refresh-ttl
- ip pim dense trace file
- ip pim dense trace flag
- show ip pim dense-mode interface-summary
- show ip pim dense-mode mrt
- show ip pim dense-mode mrt-summary
- show ip pim grafts

**Protocol Independent Multicast - Dense Mode (PIM-DM) Overview**

The PIM-DM protocol can be considered the follow-on to the DVMRP protocol. It is a "flood-and-prune" protocol. Multicast data from a source addressed to a multicast group is "flooded" to all parts of the network. Last-hop routers that detect no member of the destination group "prune" back towards the source. This type of multicast tends to be more efficient when receivers are "densely" distributed throughout the network.

The biggest difference between PIM-DM and DVMRP is that PIM-DM is only a multicast tree-building protocol whereas DVMRP is both a tree-building protocol and a network topology protocol. PIM-DM requires a unicast protocol such as OSPF to be running in the network in order to establish the underlying unicast topology.

PIM-DM namely differs from PIM-SM in that PIM-SM is an "explicit-join" protocol. An "explicit-join" grows branches of the multicast distribution trees from receivers back towards the root of the tree. In this way, multicast traffic only ever flows to members of the group. PIM-DM also differs from PIM-SM in that in PIM-DM all distribution trees are rooted at the sender. PIM-DM has no notion of shared trees, Rendezvous-Points (RPs), (*,G) state, and so on.

**Note:** PIM uses routes in the multicast Routing Information Base (RIB) to perform its RPF check. By default none of the IGPs (OSPF, RIP) place routes into the multicast RIB. They must be configured to do so. Check the relevant protocol's configuration section in order to obtain the correct syntax.

**ip pim dense-mode**

**Name**

- `ip pim dense-mode` - enables PIM dense mode on an interface

**Syntax**

- `ip pim dense-mode`
no ip pim dense-mode

**Mode**
Interface Configuration

**Parameters**
none

**Description**
The `ip pim dense-mode` command enables PIM dense mode on the associated interface.

**Default**
PIM dense mode is disabled by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the PIM dense mode protocol on interface fxp0.
```
(config)# interface fxp0
(config-if)# ip pim dense-mode
```

**ip pim graft-retry-interval**

**Name**
`ip pim graft-retry-interval` - specifies the value to use for the Retry Interval for sending Graft messages if a Graft acknowledgement has not been received

**Syntax**
```
ip pim graft-retry-interval time-seconds
no ip pim graft-retry-interval time-seconds?
```

**Mode**
Interface Configuration

**Parameters**
`time-seconds` - specifies an integer between 1 and 65535, inclusive, for the number of seconds

**Description**
The `ip pim graft-retry-interval` command specifies the value to use for the Retry Interval for sending Graft messages if a Graft Acknowledgement has not been received. The negative form of this command, `no ip pim graft-retry-interval`, removes the configured `time-seconds` value and returns it to its default. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If `ip pim graft-retry-interval` is not specified, it is the same as if the user had specified the following:
```
(config-if)# ip pim graft-retry-interval 3
```
Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures a graft retry interval of 10 seconds for interface fxp0.

```
(config)# interface fxp0
(config-if)# ip pim dense mode
(config-if)# ip pim graft-retry-interval 10
(config-if)# exit
```

**ip pim require-genid**

**Name**
ip pim require-genid - specifies whether to reject packets without a generation ID

**Syntax**

```
ip pim require-genid
no ip pim require-genid
```

**Mode**
Interface Configuration

**Parameters**

none

**Description**
Use the `ip pim require-genid` command to specify that Advanced Routing Suite should reject packets without the generation ID hello option. The negative form of this command, `no ip pim require-genid`, specifies that Advanced Routing Suite will allow such packets.

**Default**
Advanced Routing Suite accepts packets without a generation ID by default. Therefore, if `ip pim require-genid` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip pim require-genid
```

**Command History**
NGC 2.2 - This command was introduced.

**Example**
The following example specifies that Advanced Routing Suite should reject packets that do not have a generation ID.

```
(config)# interface fxp1
(config-if)# ip pim dense-mode
(config-if)# ip pim require-genid
(config-if)# exit
(config)#
```
**ip pim source-lifetime**

**Name**

`ip pim source-lifetime` - specifies the interval during which a directly attached router will continue to send State Refresh messages in the absence of additional multicast data.

**Syntax**

```
ip pim source-lifetime time-seconds
no ip pim source-lifetime time-seconds?
```

**Mode**

Global Configuration

**Parameters**

`time-seconds` - the time in seconds, specified as a non-negative integer between 4 and 32766, inclusive

**Description**

The `ip pim source-lifetime` command specifies the interval during which a directly attached router will continue to send State Refresh messages in the absence of additional multicast data. The negative form of this command, `no ip pim source-lifetime`, removes the configured value and returns this to its default value of 210 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim source-lifetime` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim source-lifetime 210
```

**Command History**

NGC 2.2 - This command was introduced.
NGC 2.3 - The lower limit of this command changed from 1 to 4.

**Examples**

The following example configures a source lifetime of 150 seconds.

```
(config)# ip pim source-lifetime 150
```

**ip pim state-refresh-capable**

**Name**

`ip pim state-refresh-capable` - specifies whether to enable the PIM-DM State Refresh option

**Syntax**

```
ip pim state-refresh-capable
no ip pim state-refresh-capable
```

**Mode**

Interface Configuration

**Parameters**

none
Description
The `ip pim state-refresh-capable` command specifies whether to enable the PIM-DM State Refresh option. Use this command to optimize network traffic by reducing the amount of flooding and pruning that occurs.

Default
The state refresh capability is enabled by default.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example disables the state refresh capability on interface eth3.

```
(config)# interface eth3
(config-if)# ip pim dense-mode
(config-if)# no ip pim state-refresh-capable
(config-if)# exit
(config)#
```

Example 2
The following example re-enables the state refresh capability on interface eth3. Note that dense mode was already configured on interface eth3 in Example 1.

```
(config)# interface eth3
(config-if)# ip pim state-refresh-capable
(config-if)# exit
(config)#
```

`ip pim state-refresh-interval`

Name
`ip pim state-refresh-interval` - specifies the number of seconds between sent State Refresh messages

Syntax
```
ip pim state-refresh-interval time-seconds
no ip pim state-refresh-interval time-seconds?
```

Mode
Global Configuration

Parameters
`time-seconds` - the time in seconds, specified as a non-negative integer between 1 and 255, inclusive

Description
The `ip pim state-refresh-interval` command specifies the frequency for sending State Refresh messages. The negative form of this command, `no ip pim state-refresh-interval`, removes the configured value and returns this to its default value of 60 seconds. **Note:** Specifying a value for `time-seconds` would be relevant in a production environment where traffic optimization is crucial.
seconds in the no form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `ip pim state-refresh-interval` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim state-refresh-interval 60
```  

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the state refresh interval to be 70 seconds.

```
(config)# ip pim state-refresh-interval 70
```  

**ip pim state-refresh-rate-limit**

Name
`ip pim state-refresh-rate-limit` - configures the minimum number of seconds that must pass before a source-group combination will accept another State Refresh message

Syntax

```
ip pim state-refresh-rate-limit time-seconds
no ip pim state-refresh-rate-limit time-seconds?
```

Mode
Global Configuration

Parameters
`time-seconds` - an integer between 0 and 254, inclusive, specifying a number of seconds

Description
Use the `ip pim state-refresh-rate-limit` command to specify the minimum length of time that must pass before another State Refresh message will be accepted for a particular source-group combination. The negative form of this command, `no ip pim state-refresh-rate-limit`, removes the configured `time-seconds` value and returns this to its default value of 30 seconds. **Note:** Specifying a value for `time-seconds` in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If `ip pim state-refresh-rate-limit` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim state-refresh-rate-limit 30
```  

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the state refresh rate limit value to be 45 seconds.

```
(config)# ip pim state-refresh-rate-limit 45
```
**ip pim state-refresh-ttl**

**Name**

`ip pim state-refresh-ttl` - configures the TTL value that will be used in state refresh messages originated by this router

**Syntax**

```
ip pim state-refresh-ttl num
no ip pim state-refresh-ttl num?
```

**Mode**

Global Configuration

**Parameters**

`num` - an integer from 1 to 255, inclusive

**Description**

Use the `ip pim state-refresh-ttl` command to configure the Time-To-Live (TTL) value that will be used in state refresh messages originated by this router. The negative form of this command, `no ip pim state-refresh-ttl`, removes the configured `num` value and returns this to its default value of 255. **Note:** Specifying a value for `num` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim state-refresh-ttl` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim state-refresh-ttl 255
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the state refresh TTL to be 100.

```
(config)# ip pim state-refresh-ttl 100
```

**ip pim dense trace file**

**Name**

`ip pim dense trace file` - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed

**Syntax**

```
ip pim dense trace file file_name [no-timestamp || overwrite]?
no ip pim dense trace file file_name [no-timestamp || overwrite]?
```

**Mode**

Global Configuration

**Parameters**

`file_name` - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.
no-timestamp - specifies that a timestamp should not be prepended to all trace lines
overwrite - specifies to begin tracing by appending or truncating an existing file

Description

The `trace file` command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. The `ip pim dense trace file` command specifies a file for tracing of all PIM-DM events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the `ip pim dense trace flag` command.

The `no-timestamp` option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The `overwrite` option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands. Consider the following example:

```
(config)# ip pim dense trace file /var/log/pimdm.log no-timestamp
(config)# ip pim dense trace file /var/log/pimdm.log
```

The option given in the second command completely replaces that given in the first.

Default

PIM-DM tracing is turned off by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, PIM-DM tracing is written to the file "/var/tmp/pimdm.log". No timestamp will display at the beginning of the trace lines.

```
(config)# ip pim dense trace file /var/tmp/pimdm.log no-timestamp
```

**ip pim dense trace flag**

Name

`ip pim dense trace flag` - specifies PIM-DM-specific tracing options as well as options that are common across all protocols

Syntax

```
ip pim dense trace flag ( [ route | normal | state | policy | task | timer | all | alert | debug ] ) |
( [ assert | graft | hello | packets | state-refresh ]
[ send | receive | send-receive ]? [detail?] )
```

```
no ip pim dense trace flag ( [ route | normal | state | policy | task | timer | all | alert | debug ] ) |
( [ assert | graft | hello | packets | state-refresh ]
[ send | receive | send-receive ]? [detail?] )
```

Mode

Global Configuration

Parameters

Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item.
```
Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

IM-specific flags that do not allow associated action items:

- `
  [ alert | debug ]` - These PIM-DM-specific flags cannot be associated with the send, receive, send-receive, or detail action items. These flags are defined as follows:
  - **alert** - trace certain failures, such as memory allocation failures, that result in a strong warning
  - **debug** - extra trace information of use mainly to developers

IM-specific flags that allow associated action items:

- `
  [ assert | graft | hello | packets | state-refresh ]` - These PIM-DM-specific flags can be associated with the send, receive, send-receive, and detail action items. These flags are defined as follows:
  - **assert** - specifies to trace PIM-DM Assert packets
  - **graft** - specifies to trace PIM-DM graft messages
  - **hello** - specifies to trace PIM-DM Hello packets
  - **packets** - specifies to trace PIM-DM packets
  - **state-refresh** - specifies to trace PIM-DM state-refresh messages

- `
  [ send | receive | send-receive ]?` - optionally specify whether to limit the tracing to packets sent, received, or both

- `[detail?]` - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

**Description**

Use the `ip pim dense trace flag` command to specify tracing flags for PIM-DM tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both assert and graft packets in the same command.

**Default**

The default is for no flags to be explicitly configured.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, trace flags specify that both the sent and received state-refresh and graft messages are traced in detail. This tracing information will be written to the file `/var/tmp/pimdm.log`.

```
(config)# ip pim dense trace file /var/tmp/pimdm.log
(config)# ip pim dense trace flag state-refresh send-receive detail
(config)# ip pim dense trace flag graft send-receive detail
```
**show ip pim dense-mode interface-summary**

**Name**
show ip pim dense-mode interface-summary - displays summarized information about interfaces currently running PIM Dense Mode

**Syntax**
```
show ip pim dense-mode interface-summary
```

**Mode**
User Execution

**Parameters**
none

**Description**
The `show ip pim dense-mode interface-summary` query displays a summary of the number of interfaces currently running PIM-DM.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example displays information returned in response to the Dense Mode Interface Summary query.
```
> show ip pim dense-mode interface-summary
PIM Dense Mode Interface Summary
Number of active interfaces 100
```

**Field Descriptions**
The following table describes the fields that appear in the PIM-DM MRT Summary Query.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of active interfaces</td>
<td>The number of active interfaces in PIM Dense Mode</td>
</tr>
</tbody>
</table>

**show ip pim dense-mode mrt**

**Name**
show ip pim dense-mode mrt - displays detailed information about S,G entries

**Syntax**
```
show ip pim dense-mode mrt [ group group-prefix || source source-prefix ]?
```
Protocol Independent Multicast (PIM)

Mode
User Execution

Parameters

group group-prefix - optionally specify a group prefix in the format a.b.c.d/e
source source-prefix - optionally specify a source prefix in the format a.b.c.d/e

Description

The show ip pim dense-mode mrt query displays detailed information about S,G entries in the Multicast Routing Table.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1
The following example displays detailed MRT information for all S,G entries present.

    > show ip pim dense-mode mrt

    224.1.1.1
    192.168.10.1 uptime 01:02:09
    Incoming interface: fxp1, index 2, vif 1
    Outgoing interface list:
    fxp2, index 3, vif 2
    fxp3, index 4, vif 3

    192.168.121.1 uptime 12:12:12
    Incoming interface: fxp2, index 3, vif 2
    Outgoing interface list:
    fxp1, index 2, vif 1

    226.2.2.2
    * uptime 02:03:24
    Incoming interface: null
    Outgoing interface list:
    fxp2, index 3, vif 2

    10.1.1.1 uptime 00:10:13
    Incoming interface: fxp3, index 4, vif 3
    Outgoing interface list:
    fxp2, index 3, vif 2

Example 2

The following query displays detailed MRT information for group prefix 226.0.0.0/24.

    > show ip pim dense-mode mrt group 226.0.0.0/24

    226.2.2.2
    * uptime 02:03:24
    Incoming interface: null
    Outgoing interface list:
    fxp2, index 3, vif 2

    10.1.1.1 uptime 00:10:13
    Incoming interface: fxp3, index 4, vif 3
    Outgoing interface list:
    fxp2, index 3, vif 2
Example 3
The following query displays detailed MRT information for group prefix 226.0.0.0/24 with source prefix 10.0.0.0/8.

```
> show ip pim dense-mode mrt group 226.0.0.0/24
  source 10.0.0.0/8
  226.2.2.2
  10.1.1.1 uptime 00:10:13
  Incoming interface: fxp3, index 4, vif 3
  Outgoing interface list:
    fxp2, index 3, vif 2
```

Field Descriptions
The following table describes the fields that appear in the PIM-DM MRT Detail Information Query (using fields in Example 3).

PIM-DM MRT Detail Information Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>226.2.2.2</td>
<td>Address of multicast group</td>
</tr>
<tr>
<td>10.1.1.1</td>
<td>IP address of source</td>
</tr>
<tr>
<td>uptime</td>
<td>Amount of time the entry has existed</td>
</tr>
<tr>
<td>fxp3</td>
<td>The physical interface name</td>
</tr>
<tr>
<td>index</td>
<td>The interface index</td>
</tr>
<tr>
<td>vif</td>
<td>The interface vif number</td>
</tr>
</tbody>
</table>

**show ip pim dense-mode mrt-summary**

**Name**
`show ip pim dense-mode mrt-summary` - displays summarized information about the PIM-DM Multicast Routing Table

**Syntax**
```
show ip pim dense-mode mrt-summary
```

**Mode**
User Execution

**Parameters**
none

**Description**
The `show ip pim dense-mode mrt-summary` query displays a summary of the number of S,G entries present in the Multicast Routing Table.
**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example displays information returned in response to the Dense Mode MRT Summary query.

```
> show ip pim dense-mode mrt-summary
PIM DM MRT Summary
Number of S,G entries 100
```

**Field Descriptions**
The following table describes the fields that appear in the PIM-DM MRT Summary Query.

PIM-DM MRT Summary Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of S,G entries</td>
<td>The number of entries in the MRT</td>
</tr>
</tbody>
</table>

**show ip pim grafts**

**Name**
show ip pim grafts - displays information about unacknowledged grafts

**Syntax**
```
show ip pim grafts
```

**Mode**
User Execution

**Parameters**
none

**Description**
The `show ip pim grafts` query displays information about unacknowledged grafts.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example information displays as a result of the `show ip pim grafts` query.

```
> show ip pim grafts
PIM Graft Information
192.168.10.1, 224.1.1.1 age 00:00:24 retransmit in 00:00:02
```

**Field Descriptions**
The following table describes the fields that appear in the PIM-DM Graft Retransmission Query.
PIM-DM Graft Retransmission Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.1</td>
<td>Address of the source</td>
</tr>
<tr>
<td>224.1.1.1</td>
<td>Address of the group</td>
</tr>
<tr>
<td>age</td>
<td>Amount of time that graft has been unacknowledged</td>
</tr>
<tr>
<td>retransmit in</td>
<td>Time left until the next transmission of graft</td>
</tr>
</tbody>
</table>
Chapter 19

Protocol Independent Multicast - Sparse Mode (PIM-SM)

In This Chapter

- Protocol Independent Multicast - Sparse Mode (PIM-SM) Overview 389
- ip pim associate-msdp 390
- ip pim bsr-admin-scope 390
- ip pim bsr-border 391
- ip pim bsr-candidate 392
- ip pim bsr-candidate global 393
- ip pim bsr-candidate group 393
- ip pim bsr-candidate interval 394
- ip pim bsr-candidate priority 395
- ip pim bsr-holdtime 396
- ip pim dr-switch-immediate 396
- ip pim mrt-spt-multiplier 397
- ip pim probe-interval 398
- ip pim register-suppression-timeout 399
- ip pim rp-address 399
- ip pim rp-candidate 400
- ip pim rp-candidate advertisement-interval 401
- ip pim rp-candidate group 402
- ip pim rp-candidate holdtime 403
- ip pim rp-candidate priority 403
- ip pim rp-switch-immediate 404
- ip pim sparse-mode 405
- ip pim threshold 406
- ip pim threshold-dr 407
- ip pim threshold-rp 407
- ip pim trace file 408
- ip pim trace flag 410
- ip pim whole-packet-checksum 411
- show ip pim bsr-router 412
- show ip pim cbsr 413
- show ip pim rp 413
- show ip pim rp-candidate 414
- show ip pim rp-hash 414
- show ip pim sparse-mode join-prune xmit 415
- show ip pim sparse-mode mrt 415

Protocol Independent Multicast - Sparse Mode (PIM-SM) Overview

PIM-SM differs from PIM-DM in that PIM-SM is an "explicit-join" protocol. An "explicit-join" grows branches of the multicast distribution trees from receivers back towards the root of the tree. In this way, multicast traffic only ever flows to members of the group. PIM-SM also differs from PIM-DM in that in PIM-DM all
distribution trees are rooted at the sender. PIM-DM has no notion of shared trees, Rendezvous-Points (RPs), (*,G) state, and so on.

You can configure PIM-SM by entering an ip pim sparse-mode command. This indicates that only PIM-SM should run on the associated interfaces.

**Note:** PIM uses routes in the multicast Routing Information Base (RIB) to perform its RPF check. By default none of the IGPs (IS-IS, OSPF, RIP) place routes into the multicast RIB. They must be configured to do so. Check the relevant protocol's configuration section in order to obtain the correct syntax.

### ip pim associate-msdp

**Name**

`ip pim associate-msdp` - informs the MSDP component of active sources in the local PIM-SM domain

**Syntax**

```
ip pim associate-msdp
no ip pim associate-msdp
```

**Mode**

Global Configuration

**Parameters**

none

**Description**

Use the `ip pim associate-msdp` command to inform the MSDP protocol of active sources in the local PIM-SM domain. The negative form of this command, `no ip pim associate-msdp`, removes this association with the MSDP protocol.

**Default**

If `ip pim associate-msdp` is not specified, it is the same as if the user had specified the following:

```
(config)# no ip pim associate-msdp
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the Associate MSDP option.

```
(config)# ip pim associate-msdp
```

### ip pim bsr-admin-scope

**Name**

`ip pim bsr-admin-scope` - configures additional rules when using the BSR RP set distribution mechanism

**Syntax**

```
ip pim bsr-admin-scope
no ip pim bsr-admin-scope
```
Mode
Global Configuration

Parameters
none

Description
This command specifies that the Admin Scope BSR rules, as defined in the IETF document draft-ietf-pim-sm-bsr-02, are to be used to distribute RP set information. By default, Advanced Routing Suite uses the single-BSR-per-domain BSR rules to distributed RP set information.

Note that it is a misconfiguration to combine these two styles of BSR in the same PIM-SM domain. If a BSR mechanism is desired for distributing RP set information, then all routers must be configured with ip pim bsr-admin-scope. If not, then none of them may be configured with ip pim bsr-admin-scope. Furthermore, when reconfiguring a router from using the Admin Scope BSR rules to using the single-BSR-per-domain BSR rules, or vice versa, all BSR state on the router is reset, which can result in a disruption of multicast traffic.

Default
If ip pim bsr-admin-scope is not specified, it is the same as if the user had specified the following:

```
(config)# no ip pim bsr-admin-scope
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example specifies to use Admin Scope BSR rules when using the BSR RP set distribution mechanism.

```
(config)# ip pim bsr-admin-scope
```

**ip pim bsr-border**

Name
ip pim bsr-border - configures a PIM boundary on the associated interface

Syntax
```
ip pim bsr-border
no ip pim bsr-border
```

Mode
Interface Configuration

Parameters
none

Description
The ip pim bsr-border command configures a PIM boundary on the associated interface. A PIM boundary prevents the transmission and receipt of Bootstrap Router (BSR) messages. This option is useful when joining two administratively separate multicast domains via Multicast Source Discover Protocol (MSDP).

The negative form of this command, no ip pim bsr-border, remove any existing PIM border configured on the interface.
Default
If `ip pim bsr-border` is not specified, it is the same as if the user had specified the following:

```
(config-if)# no ip pim bsr-border
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the PIM sparse mode protocol on interface fxp1. A PIM border is also configured on this interface.

```
(config)# interface fxp1
(config-if)# ip pim sparse-mode
(config-if)# ip pim bsr-border
(config-if)# exit
(config)#
```

**ip pim bsr-candidate**

**Name**
`ip pim bsr-candidate` - configures a candidate bootstrap router on a specified interface

**Syntax**
```
ip pim bsr-candidate interface
no ip pim bsr-candidate interface?
```

**Mode**
Global Configuration

**Parameters**
`interface` - a physical interface name or a valid IPv4 address for the BSR message origination

**Description**
The `ip pim bsr-candidate` command configures the router to be a Candidate Bootstrap Router (CBSR) on the specified interface. Only one interface can be configured as a CBSR. If any of the BSR Candidate commands are given, then candidate BSR mechanism is enabled. If an interface is not specified, then the first PIM-SM enabled interface is selected.

The negative form of this command, `no ip pim bsr-candidate`, removes the configured interface. **Note:** Specifying a value for `interface` in the `no` form has no effect on the configuration. It will not remove other BSR candidate configurations, such as priority or interval. Thus, it is displayed above as optional.

**Default**
A router does not act as a CBSR by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the router to be a CBSR on interface fxp0.

```
(config)# ip pim bsr-candidate fxp0
```
(config)#

**ip pim bsr-candidate global**

**Name**

`ip pim bsr-candidate global` - configures the router to be a CBSR in the global scope range

**Syntax**

```
ip pim bsr-candidate global [priority level]?
no ip pim bsr-candidate global [priority level]?
```

**Mode**

Global Configuration

**Parameters**

`priority level` - optionally specify a priority to be associated with the router volunteering to be a Candidate Bootstrap Router (CBSR) on the given interface. The value for level can be an integer from 0 to 255, inclusive.

**Description**

The `ip pim bsr-candidate global` command configures a router to be a CSBR in global scope range for the new ASBSR implementation syntax (for group ranges not within 239/8). You can specify an optional priority to be associated with this router. If this command is not specified, then PIM reverts to the older ASBSR implementation mechanism.

**Default**

If `ip pim bsr-candidate global` is not specified, it is the same as if the user had specified the following:

```
(config)# no ip pim bsr-candidate global
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the router to be a CBSR in global scope, with a priority of 1, on interface fxp1.

```
(config)# ip pim bsr-candidate fxp1
(config)# ip pim bsr-candidate global priority 1
(config)#
```

**ip pim bsr-candidate group**

**Name**

`ip pim bsr-candidate group` - specifies a set of multicast groups and optional priorities for which the router will volunteer to be a Candidate Bootstrap Router (CBSR)

**Syntax**

```
ip pim bsr-candidate group prefix [priority level]?
nor ip pim bsr-candidate group prefix [priority level]?
```
Mode
Global Configuration

Parameters

prefix – a valid IPv4 prefix specified as a.b.c.d/e

priority level – specifies a priority to be associated with the groups for which a router is volunteering to be a Candidate Bootstrap Router (CBSR) on the given interface. The value of level can be an integer from 0 to 255, inclusive.

Description

Use the `ip pim bsr-candidate group` command to configure the administratively scoped BSR mechanism. The admin scope prefix is 239/8. The group range, therefore, can only be within 239/8. In addition, you can optionally specify a priority to associate with the group being configured.

Default

A router does not act as a CBSR for a group by default.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the router to be a CBSR in the multicast group 239.0.0.0/8. The priority is configured to be 10.

```
(config)# ip pim bsr-candidate group 239.0.0.0/8 priority 10
```

**ip pim bsr-candidate interval**

Name

`ip pim bsr-candidate interval` – specifies the interval between originating bootstrap messages

Syntax

```
ip pim bsr-candidate interval time-seconds
no ip pim bsr-candidate interval time-seconds?
```

Mode

Global Configuration

Parameters

`time-seconds` – an integer between 1 and 536,870,906, inclusive, specifying a number of seconds

Description

If a router is acting as the elected BSR for a PIM-SM domain, then the `ip pim bsr-candidate interval` command specifies the number of seconds the router should wait between successive bootstrap message transmissions. The negative form of this command, `no ip pim bsr-candidate interval`, removes the configured `time-seconds` value and returns this to its default value of 60 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

If a router is acting as the elected BSR, and `ip pim bsr-candidate interval` is not specified, then it is the same as if the user had specified the following:
(config)# ip pim bsr-candidate interval 60

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, a router is configured to act as a BSR on interface 239.0.0.4. The interval at which the router should wait between bootstrap message transmissions is then configured to be 30 seconds.

```plaintext
(config)# ip pim bsr-candidate 239.0.0.4
(config)# ip pim bsr-candidate interval 30
```

### ip pim bsr-candidate priority

**Name**

`ip pim bsr-candidate priority` - specifies the priority to use when configuring a router to be a Candidate Bootstrap Router (CBSR) on an interface or group

**Syntax**

```plaintext
ip pim bsr-candidate priority level
no ip pim bsr-candidate priority level?
```

**Mode**

Global Configuration

**Parameters**

`level` - an integer from 0 to 255, inclusive

**Description**

Use the `ip pim bsr-candidate priority` command to specify a priority to be associated with the interface or groups for which a router is volunteering to be a CBSR. When more than one router attached to a network attempts to become the CBSR, the one with the highest priority wins. (Note that this is different than the CRP priority.)

The negative form of this command, `no ip pim bsr-candidate priority`, removes the configured `level` and returns this to its default value of 0. **Note:** Specifying a value for `level` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If a router is acting as the elected BSR, and `ip pim bsr-candidate priority` is not specified, then it is the same as if the user had specified the following:

```plaintext
(config)# ip pim bsr-candidate priority 0
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, a router is configured to act as a BSR on interface 239.0.0.4. The priority is then configured to be 5.

```plaintext
(config)# ip pim bsr-candidate 239.0.0.4
(config)# ip pim bsr-candidate priority 5
```
**ip pim bsr-holdtime**

**Name**

`ip pim bsr-holdtime` - specifies the time after which the elected Bootstrap Router (BSR) will be assumed unreachable when bootstrap messages are not received from it.

**Syntax**

```
ip pim bsr-holdtime time-seconds
no ip pim bsr-holdtime time-seconds?
```

**Mode**

Global Configuration

**Parameters**

- `time-seconds` - specifies the time, in seconds, between 12 and 1,073,741,823, inclusive.

**Description**

Use the `ip pim bsr-holdtime` command to configure the length of time that must pass without a BSR message from the elected BSR before a PIM-SM router considers the elected BSR to be dead. The negative form of this command, `no ip pim bsr-holdtime`, removes the configured `time-seconds` value and returns this to its default value of 130 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim bsr-holdtime` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim bsr-holdtime 130
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the BSR holdtime to be 50 seconds.

```
(config)# ip pim bsr-holdtime 50
```

**ip pim dr-switch-immediate**

**Name**

`ip pim dr-switch-immediate` - causes a Designated Router (DR) to initiate a switch to the Shortest Path Tree (SPT) for (S,G) upon receipt of the first data packet from source S.

**Syntax**

```
ip pim dr-switch-immediate
no ip pim dr-switch-immediate
```

**Mode**

Global Configuration

**Parameters**

none
Description
The `ip pim dr-switch-immediate` command configures a Designated Router (DR) to initiate a switch to the shortest path tree upon receipt of the first data packet received from source S addressed to group G. The negative form of this command sets the value to the default, which is to switch when the traffic rate exceeds a threshold.
If this option is set, than any other options related to the default method of switching will have no effect. Once this option has been configured, the only way to return to the default mode is to deconfigure it using the `no` form.

Default
If `ip pim dr-switch-immediate` is not specified, it is the same as if the user had specified the following:

```
(config)# no ip pim dr-switch-immediate
```

Command History
NGC 2.2 – This command was introduced.

Examples
In the following example, if the router is a DR, then the first time that the router receives a Register message encapsulating data from source S addressed to group G, it will initiate a switch to the SP tree rooted at S.

```
(config)# ip pim dr-switch-immediate
```

See Also
"ip pim rp-switch-immediate" ("ip pim rp-switch-immediate" on page 404)

**ip pim mrt-spt-multiplier**

Name
`ip pim mrt-spt-multiplier` – together with the `mrt-period`, specifies the interval at which the data rate threshold for all (S,G) entries will be checked for a possible switch to the SP tree.

Syntax
```
ip pim mrt-spt-multiplier number
no ip pim mrt-spt-multiplier number?
```

Mode
Global Configuration

Parameters
`number` – an integer between 1 and 100, inclusive

Description
The `mrt-spt-multiplier` value, multiplied by the `mrt-period` value, specifies the interval at which the data rate threshold for all (S,G) entries will be checked for a possible switch to the shortest path tree. If either the `rp-switch-immediate` or `dr-switch-immediate` option is configured, then configuring this value will have no effect.

The negative form of this command, `no ip pim mrt-spt-multiplier`, resets the MRT Stale Multiplier to its default. **Note:** Specifying a value for `number` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.
Default

If `ip pim mrt-spt-multiplier` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim mrt-spt-multiplier 4
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures the MRT SPT Mult value to be 20.

```
(config)# ip pim mrt-spt-multiplier 20
```

See Also

"ip pim dr-switch-immediate" ("ip pim dr-switch-immediate" on page 396)
"ip pim rp-switch-immediate" ("ip pim rp-switch-immediate" on page 404)

**ip pim probe-interval**

Name

`ip pim probe-interval` - specifies the number of seconds prior to the RegisterStop timer expiry to send a null Register message to the Rendezvous Point (RP)

Syntax

```
ip pim probe-interval time-seconds
no ip pim probe-interval time-seconds?
```

Mode

Global Configuration

Parameters

`time-seconds` - specifies the time, in seconds, between 0 and "ip pim register-suppression-timeout" ("ip pim register-suppression-timeout" on page 399), inclusive

Description

When PIM null Register messages are used, `ip pim probe-interval` specifies the number of seconds prior to the RegisterStop timer expiry to send a null Register message to the Rendezvous Point (RP). If a PIM RegisterStop message is received from the RP before the RegisterStop timer expires, the RegisterStop timer is reset, and the sending of encapsulating Register messages is delayed.

The negative form of this command, `no ip pim probe-interval`, returns this to its default value of 5 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip pim probe-interval` is not explicitly configured, it is the same as if the user had specified the following:

```
(config)# ip pim probe-interval 5
```

Command History

NGC 2.2 - This command was introduced.
Examples
The following example configures the probe interval to be 6 seconds.

(config)# ip pim probe-interval 6

**ip pim register-suppression-timeout**

**Name**

*ip pim register-suppression-timeout* - specifies the time, in seconds, between receiving a PIM RegisterStop message and allowing Register messages encapsulating multicast data to again be sent.

**Syntax**

```plaintext
ip pim register-suppression-timeout time-seconds
no ip pim register-suppression-timeout time-seconds?
```

**Mode**

Global Configuration

**Parameters**

*time-seconds* - specifies the time, in seconds, between 1 and 3600, inclusive.

**Description**

When a router receives a RegisterStop message from a Rendezvous Point (RP) for an (S,G) pair, it must stop sending multicast data encapsulated in Register messages for some period of time. Such a router is said to be "register-suppressed" for the (S,G) pair. The *ip pim register-suppression-timeout* command specifies the number of seconds for which the router remains register-suppressed. A lower value means that the RP receives more frequent bursts of encapsulated multicast data, while a higher value means a longer join latency for new receivers. (Note that if null Registers are sent every "ip pim probe-interval" ("ip pim probe-interval" on page 398) seconds before the timeout, then Register bursts are prevented, and register-suppression-timeout can then be lowered to decrease join latency.)

The negative form of this command, *no ip pim register-suppression-timeout*, returns this to its default value of 60 seconds. **Note**: Specifying a value for *time-seconds* in the *no* form of this command has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If *ip pim register-suppression-timeout* is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim register-suppression-timeout 60
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the PIM sparse mode protocol on interface fxp1. The Register Suppression Timeout is configured to be 70 seconds.

```
(config)# ip pim register-suppression-timeout 70
```

**ip pim rp-address**

**Name**

*ip pim rp-address* - configures the router to be a Static Rendezvous Point (static RP)
Protocol Independent Multicast (PIM)

Syntax

```
ip pim rp-address address group prefix
no ip pim rp-address address group prefix
```

Mode

Global Configuration

Parameters

- `address` - a valid IPv4 address
- `group prefix` - a valid IPv4 multicast group range

Description

Advanced Routing Suite provides two mutually exclusive methods for RP set distribution: statically configured RPs and Bootstrap Router (BSR). The `ip pim rp-address` command lets you statically configure an RP set. Multiple `ip pim rp-address` commands can be used to add elements to the set.

Defaults

A static RP address is not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, RP 192.168.10.4 is configured to serve multicast group 224.40.2.1, while 192.168.10.10 is configured to serve all other multicast groups. RP 192.168.10.4 is then de-configured, causing 192.168.10.10 to serve all groups.

```
(config)# ip pim rp-address 192.168.10.4 group 224.40.2.1/32
(config)# ip pim rp-address 192.168.10.10 group 224.0.0.0/4
(config)# no ip pim rp-address 192.168.10.4 group 224.40.2.1/32
(config)#
```

**ip pim rp-candidate**

Name

`ip pim rp-candidate` - configures the router to be a Candidate Rendezvous Point (CRP) on a specified interface

Syntax

```
ip pim rp-candidate interface
no ip pim rp-candidate interface?
```

Mode

Global Configuration

Parameters

- `interface` - a physical interface name or a valid IPv4 address
Description
The `ip pim rp-candidate` command configures a router to be a Candidate Rendezvous Point (CRP). Only one interface can be configured as the CRP. A router is chosen as the RP for a multicast group from the set of CRPs via a well-known hash function. A CRP’s suitability for a given multicast group may be biased with a priority. When choosing an RP for a group from the set of CRPs, the hash function is computed for each member of the set of CRPs with the lowest priority for the group. The CRP yielding the highest hash value is selected as the RP for the group.

The negative form of this command, `no ip pim rp-candidate`, removes the configured interface. **Note:** Specifying a value for `interface` in the `no` form has no effect on the configuration. It will not remove other RP candidate configurations, such as priority or advertisement interval. Thus, it is displayed above as optional.

Default
A router does not act as a CRP on an interface by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the router to be a CRP on interface fxp0.
```
(config)# ip pim rp-candidate fxp0
(config)#
```

`ip pim rp-candidate advertisement-interval` **ip pim rp-candidate advertisement-interval**

**Name**
`ip pim rp-candidate advertisement-interval` - sets the interval at which a Candidate Rendezvous Point (CRP) will send CRP Advertisement (Adv) messages to the Bootstrap Router (BSR)

**Syntax**
```
ip pim rp-candidate advertisement-interval time-seconds
no ip pim rp-candidate advertisement-interval time-seconds?
```

**Mode**
Global Configuration

**Parameters**
`time-seconds` - an integer between 1 and 65535, inclusive, specifying a number of seconds

**Description**
When using the BSR mechanism to distribute RP set information throughout a PIM-SM domain, CRPs must periodically send C-RP-Adv messages to the BSR. The `ip pim rp-candidate advertisement-interval` command is used to specify the time, in seconds, between successive C-RP-Adv messages.

The negative form of this command, `no ip pim rp-candidate advertisement-interval`, removes the configured `time-seconds` value and returns this to its default value of 60 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If a router is acting as the elected CRP, and `ip pim rp-candidate advertisement-interval` is not specified, then it is the same as if the user had specified the following:
```
(config)# ip pim rp-candidate advertisement-interval 60
```
Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, a router is configured to act as a CRP on interface 192.168.3.20. The interval at which the router will send C-RP-Adv messages to the BSR is then configured to be 30 seconds.

```
(config)# ip pim rp-candidate 192.168.3.20
(config)# ip pim rp-candidate advertisement-interval 30
(config)#
```

**ip pim rp-candidate group**

Name
ip pim rp-candidate group - specifies a set of multicast groups and optional priorities for which the router will volunteer to be a Candidate Rendezvous Point (CRP)

Syntax
```
ip pim rp-candidate group prefix/len [priority level]?
no ip pim rp-candidate group prefix/len [priority level]?
```

Mode
Global Configuration

Parameters
prefix/len - a valid IPv4 prefix and mask length within 224.0.0.0/4

priority level - specifies a priority to be associated with the groups for which a router is volunteering to be a Candidate Bootstrap Router (CBSR) on the given interface. The value of level can be an integer from 0 to 255, inclusive.

Description
Use the `ip pim rp-candidate group` command to configure a set of multicast groups and optional priorities for which the router will volunteer to be a CRP. To configure a CRP to volunteer for multiple group ranges, issue the command multiple times, giving different ranges. In addition, you can optionally specify a priority to associate with the group being configured.

Default
A router does not act as a CRP for a group by default.

Command History
NGC 2.2 - This command was introduced.

Examples

**Example 1**
The following example configures the router to be a CRP in the multicast group 224.0.1.0/24. The priority is configured to be 10.

```
(config)# ip pim rp-candidate group 224.0.1.0/24 priority 10
```

**Example 2**
The following example configures the router to be a CRP in the multicast group 224.1.0.1/32.
(config)# ip pim rp-candidate group 224.1.0.1/32
(config)#

**ip pim rp-candidate holdtime**

**Name**

`ip pim rp-candidate holdtime` - specifies the holdtime, in seconds, advertised in Candidate Rendezvous Point Advertisement (C-RP-Adv) messages

**Syntax**

```
ip pim rp-candidate holdtime time-seconds
no ip pim rp-candidate holdtime time-seconds?
```

**Mode**

Global Configuration

**Parameters**

- `time-seconds` - specifies the time, in seconds, between 1 and 65535, inclusive

**Description**

For CRPs, `ip pim rp-candidate holdtime` specifies the holdtime advertised in C-RP-Adv messages and is used by the Bootstrap Router (BSR) to time out RPs. The negative form of this command, `no ip pim rp-candidate holdtime`, removes the configured `time-seconds` value and returns this to its default value of 150 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Defaults**

If `ip pim rp-candidate holdtime` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim rp-candidate holdtime 150
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, a router is configured to act as a CRP on interface 192.168.3.20. The holdtime is then configured to be 100 seconds.

```
(config)# ip pim rp-candidate 192.168.3.20
(config)# ip pim rp-candidate holdtime 100
(config)#
```

**ip pim rp-candidate priority**

**Name**

`ip pim rp-candidate priority` - specifies the priority to use when configuring a router to be a Candidate Rendezvous Point (CRP) on an interface or group

**Syntax**

```
ip pim rp-candidate priority level
no ip pim rp-candidate priority level?
```
Mode
Global Configuration

Parameters
level - an integer from 0 to 255, inclusive

Description
When using the BSR method of RP set distribution, CRPs will periodically send C-RP-Adv messages to the BSR. These messages specify a set of groups for which the CRP is volunteering to be an RP. In addition, a priority is associated with each set of groups and is used to decide which CRP will actually serve as the RP for a given group. When more than one router attached to a network attempts to become the CRP, the one with the lowest priority wins. (Note that this is different than the BSR priority.)

The negative form of this command, no ip pim rp candidate priority, removes the configured level and returns this to its default value of 0. Note: Specifying a value for level in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If a router is acting as the elected CRP, and is not specified, then it is the same as if the user had specified the following:

```
(config)# ip pim rp candidate priority 0
```

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, a router is configured to act as a CRP on interface 192.168.3.20. The priority is then configured to be 5.

```
(config)# ip pim rp candidate 192.168.3.20
(config)# ip pim rp candidate priority 5
(config)#
```

**ip pim rp-switch-immediate**

Name
ip pim rp-switch-immediate - causes a Rendezvous Point (RP) to initiate a switch to the Shortest Path (SP) tree for (S,G) upon receipt of the first Register message encapsulating data from source S

Syntax
```
ip pim rp-switch-immediate
no ip pim rp-switch-immediate
```

Mode
Global Configuration

Parameters
none

Description
The PIM-SM protocol allows an RP or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within Advanced Routing Suite for deciding when an SP tree switch should be initiated. One of these methods is to
initiate a switch to the SP tree for an (S,G) pair upon receipt of the first Register message containing data from source S addressed to group G.

If the `ip pim rp-switch-immediate` command is not configured, then an active RP will initiate a switch to the SP tree when the traffic rate exceeds a threshold.

If this option is set, then any other options related to the default method of switching will have no effect. Once this option is configured, the only way to return to the default mode is to deconfigure it using the `no` form.

**Defaults**
The default is to switch when the traffic rate exceeds a threshold. Thus, if `ip pim rp-switch-immediate` is not specified, it is the same as if the user had specified the following:

```
(config)# no ip pim rp-switch-immediate
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
In the following example, if this router is an RP for G, then the first time that the router receives a Register message encapsulating data from source S addressed to group G, it will initiate a switch to the SP tree rooted at S.

```
(config)# ip pim rp-switch-immediate
```

**See Also**
"ip pim dr-switch-immediate" ("ip pim dr-switch-immediate" on page 396)

---

**ip pim sparse-mode**

**Name**
ip pim sparse-mode - enables PIM sparse mode on an interface

**Syntax**
```
ip pim sparse-mode
no ip pim sparse-mode
```

**Mode**
Interface Configuration

**Parameters**
none

**Description**
The `ip pim sparse-mode` command enables PIM sparse mode on the associated interface. The negative form of this command, `no ip pim sparse-mode`, disables PIM-SM on the interface.

**Default**
PIM sparse mode is disabled by default.

**Command History**
NGC 2.2 - This command was introduced.
Examples
The following example configures the PIM sparse mode protocol on interface fxp0.

```
(config)# interface fxp0
(config-if)# ip pim sparse-mode
(config-if)# exit
(config)#
```

**ip pim threshold**

**Name**

`ip pim threshold` - specifies the threshold, in bytes per second, which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree

**Syntax**

```
ip pim threshold bps
no ip pim threshold bps?
```

**Mode**

Global Configuration

**Parameters**

`bps` - an integer between 1 and 4,294,967,295, inclusive

**Description**

The `ip pim threshold` command configures the threshold, in bytes per second, which, when exceeded for an (S,G) pair, triggers a switch to the shortest path tree rooted at S. If either the `rp-switch-immediate` or `dr-switch-immediate` option is configured, then configuring this option has no effect.

The negative form of this command returns the value to the default of 1000 bps. **Note:** Specifying a value for `bps` in the **no** form of this command has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim threshold` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim threshold 1000
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the threshold to be 2000.

```
(config)# ip pim threshold 2000
```

**See Also**

"ip pim dr-switch-immediate" ("ip pim dr-switch-immediate" on page 396)
"ip pim rp-switch-immediate" ("ip pim rp-switch-immediate" on page 404)
"ip pim threshold-dr" ("ip pim threshold-dr" on page 407)
"ip pim threshold-rp" ("ip pim threshold-rp" on page 407)
**ip pim threshold-dr**

**Name**

`ip pim threshold-dr` - specifies the threshold, in bytes per second, for a Designated Router (DR), which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree.

**Syntax**

```
ip pim threshold-dr bps
no ip pim threshold-dr bps?
```

**Mode**

Global Configuration

**Parameters**

- `bps` - an integer between 1 and 4,294,967,295, inclusive

**Description**

The `ip pim threshold-dr` command specifies the threshold, in bytes per second, for a Designated Router (DR), which, when exceeded for an (S,G) pair, triggers a switch to the shortest path tree. If either the `rp-switch-immediate` or `dr-switch-immediate` option is configured, then configuring this option has no effect.

The negative form of this command returns the value to the default of 1000 bps. **Note:** Specifying a value for `bps` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `ip pim threshold-dr` is not specified, it is the same as if the user had specified the following:

```
(config)# ip pim threshold-dr 1000
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if this router is a DR for the pair (S,G), then the data from S addressed to G must exceed an average of 2000 bytes per second before an SPT switch is initiated.

```
(config)# ip pim threshold-dr 2000
```

**See Also**

"ip pim dr-switch-immediate" ("ip pim dr-switch-immediate" on page 396)
"ip pim rp-switch-immediate" ("ip pim rp-switch-immediate" on page 404)
"ip pim threshold" ("ip pim threshold" on page 406)
"ip pim threshold-rp" ("ip pim threshold-rp" on page 407)

**ip pim threshold-rp**

**Name**

`ip pim threshold-rp` - specifies the threshold, in bytes per second, for a Rendezvous Point (RP), which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree.
Syntax

    ip pim threshold-rp bps
    no ip pim threshold-rp bps?

Mode

Global Configuration

Parameters

    bps  -  an integer between 1 and 4,294,967,295, inclusive

Description

The `ip pim threshold-rp` command specifies the threshold, in bytes per second, for a Rendezvous Point (RP), which, when exceeded for an (S,G) pair, triggers a switch to the shortest path tree. If either the `rp-switch-immediate` or `dr-switch-immediate` option is configured, then configuring this option has no effect.

The negative form of this command returns the value to the default of 1000 bps. **Note:** Specifying a value for `bps` in the `no` form of this command has no effect on the configuration. Thus, it is displayed above as optional.

Default

If `ip pim threshold-rp` is not specified, it is the same as if the user had specified the following:

    (config)# ip pim threshold-rp 1000

Command History

NGC 2.2 - This command was introduced.

Examples

The following example, if this router is an RP for G, then the data from S addressed to G must exceed an average of 2500 bytes per second before an SPT switch is initiated. If this router is a DR for the pair (S,G), then the same data must exceed an average of 2000 bytes per second before an SPT switch is initiated. The period over which the average will be calculated will be the `mrt-interval` times the `mrt-spt-multiplier`, or 60 seconds.

    (config)# ip pim threshold-rp 2500
    (config)# ip pim threshold-dr 2000
    (config)# ip pim mrt-interval 30
    (config)# ip pim mrt-spt-multiplier 2

See Also

"ip pim dr-switch-immediate" ("ip pim dr-switch-immediate" on page 396)
"ip pim rp-switch-immediate" ("ip pim rp-switch-immediate" on page 404)
"ip pim threshold" ("ip pim threshold" on page 406)
"ip pim threshold-dr" ("ip pim threshold-dr" on page 407)

**ip pim trace file**

Name

    ip pim trace file - specifies the file to receive tracing information, the size of the file, whether to overwrite existing files, and the maximum number of files allowed
Syntax

    ip pim trace file file_name [max-size file_size[M | m | K | k]? || max-files num_files || no-timestamp || overwrite]?

    no ip pim trace file file_name [max-size file_size[M | m | K | k]? || max-files num_files || no-timestamp || overwrite]?

Mode

Global Configuration

Parameters

file_name - specifies the name of the file to receive the tracing information. Note that the file name is not specified in quotes.

max-size file_size[M | m | K | k] - specifies the maximum file size in bytes (by default) or megabytes or kilobytes. The acceptable value range is 10K to 4,294,967,295 bytes. Notice that there is no space between the file_size parameter and the unit type.

max-files num_files - specifies the maximum number of files allowed in the directory. The acceptable value range is 2 to 4,294,967,295, with a default value of 4,294,967,295 files.

no-timestamp - specifies that a timestamp should not be prepended to all trace lines

overwrite - specifies that a timestamp should not be prepended to all trace lines

Description

The trace file command is associated with each protocol, so that information pertaining to a single protocol can be written to its own file. The ip pim trace file command specifies a file for tracing of all PIM events. The negative form of this command disables this tracing. The specific events that are traced are controlled by the ip pim trace flag command.

The max-size option sets a maximum on the size of the trace file. The size can be specified in megabytes by giving 'M' or 'm' (without a leading space), or it can be specified in kilobytes by giving 'K' or 'k'. If no units are given, the size is assumed to be in bytes. The range of this parameter is 10K to 2^32-1 bytes (about 4294M). When the maximum file size is reached, the file is closed and renamed to fname.0, then fname.1, and so on, until the maximum number of files specified by the max-files option is reached. The default is unlimited. The size of a trace file is limited by the file system on which the trace file resides.

The max-files option specifies the maximum number of files allowed in the specified directory. The range of this parameter is 2 to 4,294,967,295. The default is 4,294,967,295 files.

The no-timestamp option disables the pre-pending of a timestamp to all lines written to the trace file. The default is to prepend a timestamp to all lines written to a trace file.

The overwrite option specifies whether to start tracing by truncating or appending to an existing file.

Note: These options are not cumulative across multiple commands. Consider the following example:

    (config)# ip pim trace file /var/log/pim.log no-timestamp
    (config)# ip pim trace file /var/log/pim.log max-files 10

The option given in the second command completely replaces that given in the first. In order to specify both no-timestamp and max-files 10, they must be entered on the same line as follows.

    (config)# ip pim trace file /var/log/pim.log max-files 10 no-timestamp

Default

PIM tracing is turned off by default.

Command History

NGC 2.2 - This command was introduced.
Example

In the following example, PIM tracing is written to the file "/var/tmp/pim.log". The maximum size of the file is configured to be 1024 KB, and the maximum number of files permitted is 3. When the file reaches 1 megabyte in size (the maximum size in this configuration), the file is renamed to pim.log.0, and pim.log is re-created. When the next pim.log file reaches 1 megabyte in size, pim.log.0 is renamed pim.log.1, pim.log is renamed pim.log.0, and pim.log is re-created. This continues until four log files exist (the maximum allowed in this configuration). No timestamp will display at the beginning of the trace lines.

```
(config)# ip pim trace file /var/tmp/pim.log max-size 1024k max-files 4 no-timeStamp
```

**ip pim trace flag**

**Name**

`ip pim trace flag` - specifies PIM-specific tracing options as well as options that are common across all protocols.

**Syntax**

```
ip pim trace flag ( [ route | normal | state | policy | task | timer | all | debug ] ) | ( [ packets | assert | bsr | hello | register | jp ] [ send | receive | send-receive ]? [ detail? ] )

no ip pim trace flag ( [ route | normal | state | policy | task | timer | all ] ) | ( [ packets | assert | bsr | hello | register | debug | jp ] [ send | receive | send-receive ]? [ detail? ] )
```

**Mode**

Global Configuration

**Parameters**

Flags common to all protocols:

```
[ route | normal | state | policy | task | timer | all ] - These tracing flags are common to all protocols. They cannot be associated with a send, receive, or send-receive action item. Similarly, you cannot specify to show detailed information when tracing these flags. These flags are defined as follows:
```

- **route** - trace routing table changes for routes installed by this protocol or peer
- **normal** - trace normal protocol occurrences. **Note:** Abnormal protocol occurrences are always traced.
- **state** - trace state machine transition in the protocol
- **policy** - trace the application of protocol and user-specified policy to routes being imported or exported
- **task** - trace system interface and processing associated with this protocol
- **timer** - trace timer usage by this protocol
- **all** - turns on all trace flags

PIM-specific flags that do not allow associated action items:

```
[ debug ] - This PIM-specific flag cannot be associated with the send, receive, or send-receive action items. This flag is defined as follows:
```

- **debug** - extra trace information of use mainly to developers

IM-specific flags that allow associated action items:

```
[ packets | assert | hello | bsr | register | jp ] - These PIM-specific flags can be associated with the send, receive, or send-receive action items. These flags are defined as follows:
```
- packets - specifies to trace all types of PIM packets
- assert - specifies to trace PIM Assert packets
- hello - specifies to trace PIM Hello packets
- bsr - specifies to trace BSR packets
- jp - specifies to trace Join/Prune message packets
- register - specifies to trace all Register message packets

[ send | receive | send-receive ]? - optionally specify whether to limit the tracing to packets sent, received, or both
[detail?] - optionally specify to use a more verbose format when displaying information about the contents of packets instead of one or two lines

Description
Use the `ip pim trace flag` command to specify tracing flags for PIM-SM, tracing. Each flag must reside on its own configuration line. For example, you cannot specify to trace both jp and bsr packets in the same command.

Default
The default is for no flags to be explicitly configured.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, trace flags specify that both the sent and received register and jp messages are traced in detail. This tracing information will be written to the file `/var/tmp/pim.log`.

```
(config)# ip pim trace file /var/tmp/pim.log
(config)# ip pim trace flag register send-receive detail
(config)# ip pim trace flag jp send-receive detail
```

**ip pim whole-packet-checksum**

Name
`ip pim whole-packet-checksum` - specifies that checksums in Register messages should be calculated over the entire encapsulated data packet, rather than just over the Register message header

Syntax
```
ip pim whole-packet-checksum
no ip pim whole-packet-checksum
```

Mode
Global Configuration

Parameters
none

Description
Previous versions of the PIM-SM specification had the checksum of Register messages calculated over the entire message, including encapsulated data. The latest version of the specification states that the checksum should be calculated only over the Register message header, not any encapsulated data. This is
Advanced Routing Suite’s default checksum method. Use the `ip pim whole-packet-checksum` command to specify that checksums in Register messages should be calculated according to the old method.

**Defaults**

The default is to calculate checksums over the Register message header only. Thus, if `ip pim whole-packet-checksum` is not specified, it is the same as if the user had specified the following:

`(config)# no ip pim whole-packet-checksum`

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the PIM sparse mode protocol on interface fxp1. When encapsulating data from local sources in Register messages and sending them to the Rendezvous Point on interface fxp1, the message checksum will be calculated over the entire encapsulated packet, rather than just over the Register message header.

`(config)# ip pim whole-packet-checksum`

### show ip pim bsr-router

**Name**

`show ip pim bsr-router` - displays the bootstrap router (BSR)

**Syntax**

`show ip pim bsr-router`

**Mode**

User Execution

**Parameters**

none

**Description**

Use the `show ip pim bsr-router` query to view information about the router’s BSR status.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example returns information as a result of the BSR query.

```
> show ip pim bsr-router

 PIMv2 Bootstrap Information
 This system is the Bootstrap Router (BSR)
 BSR address: 224.1.8.4
 Uptime: 00:03:53, BSR Priority: 0, Hash Mask Length: 30
 Next bootstrap message in 00:00:13
```
**show ip pim cbsr**

**Name**
show ip pim cbsr - shows this router’s candidate bootstrap router (CBSR) information

**Syntax**

```
show ip pim cbsr
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use the `show ip pim cbsr` query to obtain information about the router’s CBSR status.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example displays PIM CBSR information.

```
> show ip pim cbsr
PIMv2 Candidate Bootstrap information
CBSR address: 224.2.2.2
CBSR Priority: 0, Hash mask length: 0
CBSR State: elected
```

**show ip pim rp**

**Name**
show ip pim rp - displays rendezvous point (RP) set information

**Syntax**

```
show ip pim rp
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use the `show ip pim rp` query to obtain information about the RP set that is currently in use.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example returns information about the RP set currently in use.
> show ip pim rp
The PIM RP Set
  Group: 224.0.0.0/4
  RP: 10.1.1.1
  Uptime: 00:04:10, Expires: 00:02:20, Priority: 0

**show ip pim rp-candidate**

**Name**
show ip pim rp-candidate - displays this router’s candidate rendezvous point (CRP) information

**Syntax**
```
show ip pim rp-candidate
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use the `show ip pim rp-candidate` query to obtain information about the router’s CRP status.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example returns information about this router’s CRP status.
```
> show ip pim rp-candidate
Candidate RP information
  Candidate RP address: 10.2.2.2
  CRP Holdtime: 150 seconds
  Group: 224.2.0.0/8 Priority 0
```

**show ip pim rp-hash**

**Name**
show ip pim rp-hash - displays the rendezvous point (RP) to which a group hashes

**Syntax**
```
show ip pim rp-hash ipv4_address
```

**Mode**
User Execution

**Parameters**
ipv4_address - specify a valid IPv4 multicast group address specified in dotted-quad format

**Description**
Use the `show ip pim rp-hash` query to view the RP to which a specified multicast group hashes.
Command History
NGC 2.2 - This command was introduced.

Examples
The following example displays the RP to which address 224.2.2.8 hashes.

  > show ip pim rp-hash 224.2.2.8
  RP  10.1.2.3

*show ip pim sparse-mode join-prune xmit*

Name
show ip pim sparse-mode join-prune xmit - displays sparse mode join/prune transmission information

Syntax
  show ip pim sparse-mode join-prune xmit

Mode
User Execution

Parameters
none

Description
Use the show ip pim sparse-mode join-prune xmit query to view information about sparse mode join/prune transmissions.

Command History
NGC 2.2 - This command was introduced

Examples
The following example returns sparse mode join/prune transmission information.

  > show ip pim sparse-mode join-prune xmit
  Neighbor address: 10.1.2.3
  Via interface: fxp1
  Next message in 335 seconds
  Group: 224.3.2.1
  Join: 10.24.16.0/4    RPT

*show ip pim sparse-mode mrt*

Name
show ip pim sparse-mode mrt - displays detailed information about S,G entries

Syntax
  show ip pim sparse-mode mrt [ group group-prefix ||
  source source-prefix ]?

Mode
User Execution
Parameters

group group-prefix - optionally specify a group prefix in the format a.b.c.d/e

source source-prefix - optionally specify a source prefix in the format a.b.c.d/e

Description

The `show ip pim sparse-mode mrt` query displays detailed information about PIM-SM S,G entries in the Multicast Routing Table. The query shows all S,G entries by default, or you can specify a single group and/or source.

Command History

NGC 2.2 - This command was introduced.

Examples

The following example shows a request for all MRT information.

```plaintext
> show ip pim sparse-mode mrt
PIM Sparse Mode Multicast Routing Table

Flags: E - Entry forwarding on the RPT,
       J - Joining to the SPT,
       N - No flags set, R - RPT bit is set,
       S - SPT bit is set, W - Wildcard entry,
       X - External component interest

224.1.1.1
192.168.10.1 uptime 01:02:09, flags: S
Incoming interface: fxp1, index 2, vif 1
Outgoing interface list:
   fxp2, index 3, vif 2
   fxp3, index 4, vif 3

192.168.121.1 uptime 12:12:12, flags: S
Incoming interface: fxp2, index 3, vif 2
Outgoing interface list:
   fxp1, index 2, vif 1

226.2.2.2
* uptime 02:03:24, flags: N
Incoming interface: null
Outgoing interface list:
   fxp2, index3, vif 2

10.1.1.1 uptime 00:10:13, flags S
Incoming interface: fxp3, index 4, vif 3
Outgoing interface list:
   fxp2, index 3, vif 2
```

Field Descriptions

The following table describes the fields that appear in the PIM-SM MRT Detail Information Query:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>224.1.1.1 (and other top level IPv4 addresses)</td>
<td>Address of multicast group</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>10.1.1.1 (and other IPv4 address below multicast group address)</td>
<td>IP address of source</td>
</tr>
<tr>
<td>uptime</td>
<td>Amount of time the entry has existed</td>
</tr>
<tr>
<td>flags</td>
<td>Indicates any flags that are set.</td>
</tr>
<tr>
<td>fxp3 (and other physical interfaces)</td>
<td>The physical interface name</td>
</tr>
<tr>
<td>index</td>
<td>The interface index</td>
</tr>
<tr>
<td>vif</td>
<td>The interface vif number</td>
</tr>
</tbody>
</table>
Chapter 20

Access Lists

In This Chapter

Access Lists Overview 418
access-list 418
access-list sequence-number 420
ip access-list sequence-number 420
ip access-list standard 421
permit | deny 422
show access-list 424
show ip access-list 425

Access Lists Overview

Access lists specify a collection of permit and deny conditions that apply to prefixes. They perform a sequential lookup and return the first matched entry as the true match. Access lists are referred to by name from other configuration elements, such as route maps and distribute lists. The assigned name must be unique among all other access lists.

access-list

Name
access-list - configures an access list

Syntax
access-list list_id [seq seq_value]? [deny | permit] [ip4_addr (wildcard)?family]
no access-list list_id [seq seq_value]? [deny | permit] [ip4_addr (wildcard)?family]

Mode
Global Configuration

Parameters
list_id - specify a string of characters for this access list name
seq seq_value - this is specified as an integer between 0 and 65535, inclusive. Note: Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. (See "access-list sequence-number" on page 420) for more information on automatic generation.) Otherwise, specifying this command is optional.
deny | permit - denies or permits the specified source address
[ip4_addr (wildcard)?family] - specify a valid IPv4 address with an optional wildcard to be included in this access list
**Description**

Use the `access-list` command to configure an access list. List entries are in the order specified by their sequence number. The first match determines whether the address is permitted or denied. If no conditions match, then the policy engine declares it as a failed match.

Every entry in the Access List is associated with a permit/deny indicator. It is used to indicate to the `filter/route-map` containing the list as a match element whether the match element succeeded or not. A permit indicates a successful match whereas a deny result causes the match element to fail.

When creating an access list, by default, the end of the access list contains an implicit deny statement for everything. Further, if a wildcard is omitted from an associated IPv4 host address access list specification, 0.0.0.0 is assumed to be the wildcard.

Use the negative form of this command to delete all entries or specific entries in an access list. One way to remove a specific entry from an access list is to specify all parameters that were specified when the entry was created. Another way is to specify the sequence number of the entry, which is either automatically generated or explicitly configured.

**Default**

Access lists are not configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures an access list called "abc" that denies all prefixes matching 128.0.0.0/8.

```
(config)# access-list abc deny 128.0.0.0 0.255.255.255
```

**Example 2**

The following example configures three entries in access list number 1. The first list entry permits the single address 10.11.0.12. The second list entry denies all other addresses in 10.11/16. The third entry permits all addresses in 10/8.

```
(config)# access-list 1 permit 10.11.0.12 0.0.0.0
(config)# access-list 1 deny 10.11.0.0 0.0.255.255
(config)# access-list 1 permit 10.0.0.0 0.255.255.255
```

**Example 3**

The following example deletes the first entry configured in access list number 1, shown in Example 2.

```
(config)# no access-list 1 permit 10.11.0.12 0.0.0.0
```

**Example 4**

The following example shows another way to delete the first entry configured in access list number 1, shown in Example 2.

```
(config)# no access-list 1 seq 5
```

**See Also**

"access-list sequence-number" ("access-list sequence-number" on page 420)
"access-list sequence-number" ("access-list sequence-number" on page 420)
**access-list sequence-number**

**Name**

`access-list sequence-number` - specifies whether Advanced Routing Suite should use automatic sequence numbering when configuring access lists.

**Syntax**

```
access-list sequence-number
no access-list sequence-number
```

**Mode**

Global Configuration

**Parameters**

none

**Description**

Sequence values are generated in increments of 5, with the first sequence value generated being 5, then 10, then 15, and so on. If a sequence value is specified for an entry, and then not specified for subsequent entries, the assigned (generated) sequence values are then incremented in units of 5 based on the highest sequence number present in the list.

**Note:** Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. Otherwise, specifying this command is optional. Use the negative form of the sequence command, no `access-list sequence-number`, to turn off automatic sequence number generation.

**Default**

Sequence numbers are generated automatically by Advanced Routing Suite. Therefore, if `access-list sequence-number` is not specified, it is the same as if the user had specified the following:

```
(config)# access-list sequence-number
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures four entries in access list "abc." The first entry explicitly configures sequence number 3. The two subsequent entries will, therefore, have automatically generated sequence numbers of 8 and 13 respectively. Sequence numbering is then turned off; therefore, sequence numbers must be explicitly configured for the remainder of the access list.

```
(config)# access-list abc seq 3 permit 10.11.0.12 0.0.0.0
(config)# access-list abc deny 10.11.0.0 0.0.255.255
(config)# access-list abc permit 10.0.0.0 0.255.255.255
(config)# no access-list sequence-number
(config)# access-list abc seq 20 permit 192.168.0.0 0.0.255.255
```

**ip access-list sequence-number**

**Name**

`ip access-list sequence-number` - specifies whether Advanced Routing Suite should use automatic sequence numbering when configuring access lists in Standard ACL Configuration mode.
Syntax

ip access-list sequence-number
no ip access-list sequence-number

Mode

Global Configuration

Parameters

none

Description

Sequence values are generated in increments of 5, with the first sequence value generated being 5, then 10, then 15, and so on. If a sequence value is specified for an entry, and then not specified for subsequent entries, the assigned (generated) sequence values are then incremented in units of 5 based on the highest sequence number present in the list.

Note: Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. Otherwise, specifying this command is optional. Use the negative form of the sequence command, no ip access-list sequence-number, to turn off automatic sequence number generation.

Default

Sequence numbers are generated automatically by Advanced Routing Suite. Therefore, if ip access-list sequence-number is not specified, it is the same as if the user had specified the following:

```config
ip access-list sequence-number
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example configures four entries in access list "abc." The first entry explicitly configures sequence number 3. The two subsequent entries will, therefore, have automatically generated sequence numbers of 8 and 13 respectively. Sequence numbering is then turned off; therefore, sequence numbers must be explicitly configured for the remainder of the access list.

```config
ip access-list standard abc
(config-std-nacl)# seq 3 permit 10.11.0.12 0.0.0.0
(config-std-nacl)# deny 10.11.0.0 0.0.255.255
(config-std-nacl)# permit 10.0.0.0 0.255.255.255
(config-std-nacl)# exit
(config)# no ip access-list sequence-number
(config)# ip access-list standard abc
(config-std-nacl)# seq 20 permit 192.168.0.0 0.0.255.255
(config-std-nacl)# exit
(config)#
```

ip access-list standard

Name

ip access-list standard - enters standard configuration mode for a specified access list
Syntax

    ip access-list standard list_name

Mode
Global Configuration

Parameters

list_name - specify a string of characters for this access list name

Description
Use ip access-list standard command to enter Standard Access List Configuration mode. This configuration mode provides an alternative to configuring access lists using the access-list command in Global Configuration mode.

Default
none

Command History
NGC 2.2 - This command was introduced.

Examples
The following example enters standard access list configuration mode for the access list labeled abc.

    (config)# ip access-list standard abc
    (config-std-nacl)#

permit | deny

Name
permit | deny - specifies whether to allow or deny entries in an access list

Syntax

    [seq seq_value] [deny | permit]
    [ (ip4_addr [wildcard]?) ]
    no [seq seq_value] [deny | permit]
    [ (ip4_addr [wildcard]?) ]

Mode
Standard ACL Configuration

Parameters

seq seq_value - this is specified as an integer between 0 and 65535, inclusive. Note: Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. Otherwise, specifying this command is optional.
deny | permit - denies or permits the specified access list
(ipv4_addr [wildcard]?) - specify a valid IPv4 address with an optional wildcard to be included in this access list
**Description**

Use the above access lists commands in ACL Configuration Mode to configure an access list for the specified entry. Entries are evaluated according to their list entry sequence. The first match determines whether the address is permitted or denied. If no conditions match, then the policy engine declares it as a failed match.

Every entry in the Access List is associated with a permit/deny indicator. It is used to indicate to the filter/route-map containing the list as a match element whether the match element succeeded or not. A permit indicates a successful match whereas a deny result causes the match element to fail.

When creating an access list, by default, the end of the access list contains an implicit deny statement for everything. Further, if a wildcard is omitted from an associated IPv4 host address access list specification, then 0.0.0.0 is assumed to be the wildcard.

Use the negative form of this command to delete all entries or specific entries in an access list. One way to remove a specific entry from an access list is to specify all parameters that were specified when the entry was created. Another way is to specify the sequence number of the entry, which is either automatically generated or explicitly configured.

**Default**

Access list entries are not configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures access list "abc" with three entries. The first entry permits the single address 10.11.0.12. The second entry denies all other addresses in 10.11/16. The third entry permits all other addresses in 10/8.

```
(config)# ip access-list standard abc
(config-std-nacl)# permit 10.11.0.12 0.0.0.0
(config-std-nacl)# deny 10.11.0.0 0.0.255.255
(config-std-nacl)# permit 10.0.0.0 0.255.255.255
(config-std-nacl)# exit
(config)#
```

**Example 2**

In the following example, the first list entry from Example 1 is deleted from access list abc.

```
(config)# ip access-list standard abc
(config-std-nacl)# no permit 10.11.0.12 0.0.0.0
(config-std-nacl)# exit
(config)#
```

**Example 3**

The following example shows another way to delete the first entry configured in access list abc, shown in Example 1.

```
(config)# ip access-list standard abc
(config-std-nacl)# no seq 5
(config-std-nacl)# exit
(config)#
```
See Also
"ip access-list standard" ("ip access-list standard" on page 421)

show access-list

Name

show access-list - displays information about access lists that were configured in Global Configuration mode

Syntax

    show access-list [ detail | summary ]? [ list_id ]?

Mode

Privileged Execution

Parameters

detail | summary - optionally specify whether you want the query to respond with a more verbose format. Summary information is returned by default.

list_id - a string of characters or an integer that represents the ID of a configured access list. Specifying this is optional. Advanced Routing Suite will return information for all configured access lists if a list_id is not specified.

Description

The show access-list query displays information about all or specific access lists configured in Global Configuration mode (i.e., using the access-list command). (Use the show ip access-list command to display information about access lists configured with the ip access-list command.)

This query has two forms. If it is issued without arguments, then information about all configured access lists is returned. Alternatively, the query can be issued naming a specific access list. In this case, the reply will contain information pertaining only to the referenced access list.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following query is a request for summary information for all configured access lists.

    # show access-list
    Standard IP access-list acl1: 3 entries
    seq 3 permit 10.10.0.0, wildcard bits 0.0.255.255
    seq 8 deny 10.0.0.0, wildcard bits 0.255.255.255
    seq 13 permit 12.12.0.0, wildcard bits 0.0.255.255
    Standard IP access-list acl2: 3 entries
    seq 3 permit 10.11.0.12, wildcard bits 0.0.0.0
    seq 10 deny 10.11.0.0, wildcard bits 0.0.255.255
    seq 15 permit 10.0.0.0, wildcard bits 0.255.255.255

Example 2

The following query is a request for detail information about a specific access list named "acl1".

    # show access-list detail acl1
show ip access-list

Name

show ip access-list - displays information about access lists configured in Standard ACL Configuration mode

Syntax

    show ip access-list [ detail | summary ]? [ list_id ]?

Mode

Privileged Execution

Parameters

detail | summary - optionally specify whether you want the query to respond with a more verbose format. Summary information is returned by default.

list_id - a string of characters or an integer that represents the ID of a configured access list. Specifying this is optional. Advanced Routing Suite will return information for all configured access lists if a list_id is not specified.

Description

The show ip access-list query displays information about all or specific access lists configured in Standard ACL Configuration mode (i.e., using the ip access-list command).

This query has two forms. If it is issued without arguments, then information about all configured access lists is returned. Alternatively, the query can be issued naming a specific access list. In this case, the reply will contain information pertaining only to the referenced access list.

Command History

NGC 2.2 - This command was introduced.

Examples

Example 1

The following query is a request for summary information for all configured access lists.

    # show ip access-list
    Standard IP access-list acl1: 3 entries
    seq 3 permit 10.10.0.0, wildcard bits 0.0.255.255
    seq 8 deny 10.0.0.0, wildcard bits 0.255.255.255
    seq 13 permit 12.12.0.0, wildcard bits 0.0.255.255

Example 2

The following query is a request for detail information about a specific access list named "acllist1".

    # show ip access-list detail acllist1
Standard IP access-list acllist1:
count: 3, sequences 3 - 13
seq 3 permit 10.10.0.0, wildcard bits 0.0.255.255
seq 8 deny 10.0.0.0, wildcard bits 0.255.255.255
seq 13 permit 12.12.0.0, wildcard bits 0.0.255.255
Chapter 21

AS Paths and AS Path Lists

In This Chapter

AS Paths and AS Path Lists Overview 427
ip as-path access-list 428
ip as-path name 429
show ip as-path-access-list 430
show ip bgp paths 431

AS Paths and AS Path Lists Overview

An AS path is the autonomous systems that routing information passed through to get to a specified router. It indicates the origin of this route. The AS path is used to prevent routing loops in BGP.

You can use this routing information to prefer one path to a destination network over another. To set route preferences, configure filters to apply to AS paths when importing and exporting routes. Each autonomous system through which a route passes appends its AS number to the beginning of the AS path.

Advanced Routing Suite includes a powerful implementation of AS path regular expressions. The entire AS path regular expression must be contained within the <...> or within (...). The content is read from left to right. The left side matches the beginning of the AS path; the right side matches the end of the AS path. The alphabet (set of valid members) is the valid range of AS numbers, or more specifically, \{1 ... 65535\}. Also, Advanced Routing Suite supports the following "wildcards" or expressions that can be used to build a regular expression:

- . - (period) represents any valid member of the alphabet
- * - (asterisk) matches zero or more of the preceding element/expression
- + - (plus sign) matches one or more of the preceding element/expression
- # - (pound sign) matches zero or one occurrence of the preceding element/expression

Binary operators:

- " " (AND) – any sequence of elements and/or expressions separated by a space (" ")
- "|" (OR) – any sequence of elements and/or expressions separated by the vertical line symbol ("|")

The symbols "[]" are used to delimit a set of AS numbers. The set may be a list of AS numbers separated by a space or a range of AS numbers separated by a dash (--). If the entire list of members is prefixed with a "^", then the valid members are those not listed in the set. (Because a null string or empty string is not an instance in the alphabet, AS numbers such as [^808] will not match an empty string.) Examples of AS path regular expressions follow:

Match any single AS number as the AS path:

- "."

Match all AS paths coming from a given AS that start with 808:

- "808 ..*"

Match all paths that do not end with the given AS numbers but must have at least one AS:

- ".* [^808 809]"

Match a path that has only valid exterior AS numbers:
"[1-6451]+"

**Match** 305 808 and exactly one other AS number other than 100:
"305 808 [^100]"

**Match** 305 808 and any other AS number except 100 or no additional AS. That is, match either 305 808 as the complete path or 305 808 x, where x is any integer other than 100:
"305 808 [^100]"

**Match** either 808 or 305 with no additional AS numbers in the path:
"305|808"

**Note:** For users familiar with gated.conf or Advanced Routing Suite XML configuration, some characters that are normally used in regular expressions have special meaning in the Advanced Routing Suite CLI. Therefore, these characters are substituted by others.

- `<space>` is replaced by an underscore.
  For example, the regular expressions 808 and 804 are separated by an underscore in this command:
  (config)# ip as-path name bar 808_804 any

- `?` is replaced by the character "#".
  For example, the regular expression 808? is replaced by 808# in this command:
  (config)# ip as-path name bar 808# any

---

### ip as-path access-list

**Name**

ip as-path access-list - specifies a BGP AS_PATH regular expression to be used in filtering

**Syntax**

```plaintext
ip as-path access-list name [ ( [ permit | deny ] regexp origin ) | [ named as-path-name ] ]
no ip as-path access-list name
```

**Mode**

Global Configuration

**Parameters**

- **name** - the name of this AS Path access list specified as a string of characters
- **permit | deny** - specifies whether to allow or reject this AS Path regular expression
- **regexp** - the regular expression
- **origin** - one of either egp, igp, incomplete, or any
- **named as-path-name** - specifies the name of a configured access list

**Description**

The `ip as-path access-list` command specifies a BGP AS_PATH regular expression used in filtering. An AS path access list does nothing until it is referred to in the context of policy. This must be done within a route map and then referred to within a protocol.

The origin specifies whether the route was learned from an EGP source, an IGP source, or another source. An origin of igp indicates the route was learned from an Intra-Domain Routing Protocol and is most likely complete. An origin of egp indicates the route was learned from the EGP protocol, and the path is most likely not complete. When the route is learned from another source, an origin of incomplete is used. An origin of any can be used to match any origin.
**Default**
No AS path access lists are defined by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
The following example configures an access list named "abc" with two list entries. The first entry permits regular expression 808.* with an origin of any. The second entry denies regular expression 909.* with an origin of any.

```
(config)# ip as-path access-list abc permit 808.* any
(config)# ip as-path access-list abc deny 909.* any
```

**Example 2**
The following example configures an access list named "abc" to apply the regular expression of a configured AS path named "blah."

```
(config)# ip as-path access-list abc named blah
```

**ip as-path name**

**Name**

*ip as-path name* - specifies a BGP AS_PATH regular expression to be used in filtering

**Syntax**

```
ip as-path name name regexp origin
no ip as-path name name
```

**Mode**

Global Configuration

**Parameters**

- `name` - the name of this AS Path regular expression specified as a string of characters
- `regexp` - the regular expression
- `origin` - one of either `egp`, `igp`, `incomplete`, or `any`

**Description**
The *ip as-path name* command is used to specify a BGP AS_PATH regular expression used in filtering. An AS path list does nothing until it is referred to in the context of policy. This must be done within a route map and then referred to within a protocol.

The origin specifies whether the route was learned from an EGP source, an IGP source, or another source. An *origin* of `igp` indicates the route was learned from an Intra-Domain Routing Protocol and is most likely complete. An *origin* of `egp` indicates the route was learned from the EGP protocol, and the path is most likely not complete. When the route is learned from another source, an *origin* of `incomplete` is used. An *origin* of `any` can be used to match any *origin*.

**Default**
No AS paths are defined by default.
Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following example configures an AS path regular expression called "bar" that permits regular expression 808.* with an origin of any.

    (config)# ip as-path name bar 808.* any

Example 2
The following example configures an AS path regular expression called "bar" that permits regular expression 808.* with an origin of any. The configured AS path regular expression is then referenced in an AS Path Access List configuration.

    (config)# ip as-path name bar 808.* any
    (config)# ip as-path access-list abc permit named bar

show ip as-path-access-list

Name
show ip as-path-access-list - displays information about BGP access lists

Syntax
    show ip as-path-access-list [name]?

Mode
User Execution

Parameters
    name - optionally specify a single AS path access list name

Description
Use the show ip as-path-access-list query to obtain information about BGP access lists. If a name is given, then the information returned pertains only to that access list. If name is not given, then information about all AS path access lists is returned.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example is a request for information about a single access list called "abc."

    > show ip as-path-access-list abc
    AS path access list abc
deny .*
    permit 201

Field Descriptions
The following table describes the fields that appear in the BGP AS Path Access List Query.
BGP AS Path Access List Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS path access list</td>
<td>The name of the AS Path Access List</td>
</tr>
<tr>
<td>permit/deny</td>
<td>Specifies whether the list is permitted or denied, followed by an AS Path regular expression</td>
</tr>
</tbody>
</table>

**show ip bgp paths**

**Name**
show ip bgp paths - displays information about BGP AS Paths

**Syntax**

```plaintext
show ip bgp paths
```

**Mode**
User Execution

**Parameters**
none

**Description**
Use the `show ip bgp paths` query to obtain information about all BGP AS Paths.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example returns information for BGP AS Paths.

```plaintext
> show ip bgp paths
Refcount | Metric | Path
5        | 50     | 202 201 200
10       | 10     | 200
```

**Field Descriptions**
The following table describes the fields that appear in the BGP AS Path Query.

BGP AS Path Query Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refcount</td>
<td>The number of routes using that path</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Metric</td>
<td>The MED value for the path</td>
</tr>
<tr>
<td>Path</td>
<td>The AS Path value</td>
</tr>
</tbody>
</table>
Chapter 22

BGP Communities and Community Lists

In This Chapter

BGP Communities and Community Lists Overview 433
ip community-list 433
ip community-set 434

BGP Communities and Community Lists Overview

BGP updates carry a number of path attributes. Some of these, like the AS_PATH, are mandatory and appear in every update message sent. Others are optional, and may or may not appear in any given update. Of the optional attributes, two can be specified arbitrarily by administrators to ease configuration. These two attributes are "communities" and "extended communities." Both of these attributes operate by "coloring" routes received in updates where these attributes are present; every router keeps track of the set of communities and extended communities with which a route was learned. The particular communities (or extended communities) with which a route was learned can be used to indicate that a particular set of policies should be applied to those routes.

ip community-list

Name

ip community-list - specifies a community or group of communities used in filtering or modifying community values

Syntax

    ip community-list name [permit | deny] [comm-set commset_name] [exact]? [standard | extended]?
    no ip community-list name [permit | deny] [comm-set commset_name] [exact]? [standard | extended]?

Mode

Global Configuration

Parameters

name - a string of characters that uniquely identifies this community list
permit | deny - permits or denies the entries in the community list. When creating a community list, by default, the end of the community list contains an implicit deny statement for everything.
comm-set commset_name - specify a community set or an extended community set
exact - optionally specify whether an entire community list must be matched exactly
BGP Communities and Community Lists

**standard** - specifies that the referenced community set is a standard (non-extended) community set. **standard** and **extended** are mutually exclusive. **Note:** If neither **standard** nor **extended** is specified for a standard community list, then the default value is **standard**. If both are configured, the keyword supplied last will override the previous.

**extended** - specifies that the referenced community set is an extended (non-standard) community set. **standard** and **extended** are mutually exclusive. If both are configured, the keyword supplied last will override the previous.

### Description

The `ip community-list` command is used to specify a community, an extended community, or groups of both used in filtering or modifying community values. Multiple community lists for the same name can be specified to create multiple elements in the list. This list is generated in the order in which the commands are entered.

### Default

Community lists are not configured by default.

### Command History

NGC 2.2 - This command was introduced.

### Examples

#### Example 1

The following example configures a community list with two list entries. The first entry permits a community set labeled set1, and the second entry permits a community set labeled set2. Note that set1 and set2 are implicitly taken to refer to the names in the standard community set name space.

```
(config)# ip community-set set1 101:101
(config)# ip community-set set2 101:104
(config)# ip community-list abc permit comm-set set1 exact
(config)# ip community-list abc permit comm-set set2 exact
```

#### Example 2

The following example configures a community set, "set1", that includes AS:num 101:102. It then configures an extended community set "ext-set1", that includes Route Target AS:num 201:202. The two are then added to an extended community list, called "commlist1".

```
(config)# ip community-set set1 101:102
(config)# ip community-set ext-set1 extended rt 201:202
(config)# ip community-list commlist1 permit comm-set set1 extended
(config)# ip community-list commlist1 permit comm-set ext-set1 extended
```

### ip community-set

**Name**

`ip community-set` - specifies a community or group of communities or extended communities used in filtering or modifying community values

**Syntax**

**Standard BGP Communities**

```
ip community-set set_id [as:num | comm-num | local-as | no-advertise | no-export | none ] {1,n}
```
no ip community-set set_id [as:num | comm-num | local-as | no-advertise | no-export | none ] (1,n)

Extended BGP Communities

ip community-set set_id extended [ none | (rt [as:num | ip:num]) | (rt4 [as:num | ip:num]) | (soo [as:num | ip:num]) | (soo4 [as:num | ip:num]) | (lbw as:float) (*:* ) ] {1,n}

no ip community-set set_id extended [ none | (rt [as:num | ip:num]) | (soo [as:num | ip:num]) | (lbw as:float) (*:* ) ] {1,n}

Mode

Global Configuration

Parameters

Standard BGP Communities

set_id - a string of characters that uniquely identifies this community set

as:num - the autonomous system (AS) number to be concatenated with a number specified as a 16-bit number. This is used for specifying a community in the "community-split" style.

comm-num - the concatenation value of the two sixteen-bit numbers used for this arbitrary community

local-as - specifies the well-known community NO_EXPORT_SUBCONFED as defined in RFC 1997. Routes tagged with this community are not to be advertised to external peers, including those peers in other members’ autonomous systems inside of a BGP confederation.

no-advertise - specifies the well-known community NO_ADVERTISE as defined in RFC 1997. Routes tagged with this community are not to be advertised to any other peers.

no-export - specifies the well-known community NO_EXPORT as defined in RFC 1997. Routes tagged with this community are not to be exported outside of a confederation boundary when confederations are in use or if they are outside of the local AS when confederations are not configured.

none - specifies the empty set of communities. It is useful only where matches are being attempted. When used in that context, if a route has any communities associated with it, then it does not match; otherwise, it does match. Note: Empty sets of communities cannot be combined with other communities in the list.

{1,n} - this command must be used at least once, and can be used an unlimited number of times for a single community set ID

Extended BGP Communities

set_id - a string of characters that uniquely identifies this community set

none - specifies the empty set of extended communities. Note: Empty sets of extended communities cannot be combined with other communities in the list.

rt (as:num | ip:num) - specifies the Route Target extended community. This is followed by either the autonomous system (AS) number to be concatenated with num (specified as a 32-bit number), or an IPv4 address in dotted-quad format with a 16-bit number value for num.

rt4 (as:num | ip:num) - specifies a 4-bit Route Target extended community. This is followed by either the autonomous system (AS) number to be concatenated with num (specified as a 2-bit number), or an IPv4 address in dotted-quad format with a 16-bit number value for num.

soo (as:num | ip:num) - specifies the Site of Origin (Route Origin) extended community. This is followed by either the autonomous system (AS) number to be concatenated with num (specified as a 32-bit number), or an IPv4 address in dotted-quad format with a 16-bit number value for num.

soo4 (as:num | ip:num) - specifies the Site of Origin (Route Origin) extended community. This is followed by either the 4-bit autonomous system (AS) number to be concatenated with num (specified as a 2-bit number), or an IPv4 address in dotted-quad format with a 16-bit number value for num.
lbw as:float - specifies the Link Bandwidth extended community. This is followed by the autonomous system number and floating point numbering expressing the speed of a link in bytes per second.

*: - specifies all extended communities

{i,n} - this command must be used at least once, and can be used an unlimited number of times for a single extended community ID

**Description**

The ip community-set command is used to specify a set of communities and extended communities that must be matched for a route to be considered a match.

BGP updates carry a number of path attributes. Some of these, like the s, are mandatory and appear in every update message sent. Others are optional, and may or may not appear in any given update. Of the optional attributes, two can be specified arbitrarily by administrators to ease configuration. These two attributes are "communities" and "extended communities." Both of these attributes operate by "coloring" routes received in updates where these attributes are present; every router keeps track of the set of communities and extended communities with which a route was learned. The particular communities (or extended communities) with which a route was learned can be used to indicate that a particular set of policies should be applied to those routes.

**Notes:**

- You cannot mix standard and extended communities in the same community set.
- If you configure a community set twice, the second configuration overwrites the first. The two configurations will not merge.
- Separate name spaces are used for standard and extended community sets. Thus, the same set name can be used for both a standard and an extended community set.

**Default**

Community sets are not configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

**Example 1**

The following example configures a community set named "abc" that includes AS:num 201:202.

```
(config)# ip community-set abc permit 201:202
```

**Example 2**

The following example configures an extended community set "ext-set1", with a Route Target AS:num 201:202 and a site of origin IP number 10.1.2.3:100.

```
(config)# ip community-set ext-set1 extended rt 201:202 soo 10.1.2.3:100
```
Chapter 23

Prefix Lists and Prefix Trees

In This Chapter

Prefix Lists and Prefix Trees Overview 437
ip prefix-list 437
ip prefix-list sequence-number 438
ip prefix-tree 439
show ip prefix-list 441
show ip prefix-tree 442

Prefix Lists and Prefix Trees Overview

Advanced Routing Suite provides two types of prefix based lists: Prefix Lists and Prefix Trees. Prefix Trees return the longest match as the true match whereas Prefix Lists simulate a sequential lookup and return the first matched entry as the true match. The entries are ordered according to entry-seq.

ip prefix-list

Name

ip prefix-list - creates a prefix list

Syntax

```
ip prefix-list list_id [seq seq_value]?
[deny | permit] [network/masklen] [ge length]? [le length]?
no ip prefix-list list_id [seq seq_value]?
[deny | permit] [network/masklen] [ge length]? [le length]?
```

Mode

Global Configuration

Parameters

**list_id** - a string of characters or an integer

**seq seq_value** - this is specified as an integer between 0 and 4,294,967,295, inclusive. **Note:** Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. Otherwise, specifying this command is optional.

deny | permit - denies or permits the specified source address

**network/ masklen** - specify a valid IPv4 address with a mask length. The IPv4 address is specified in dotted-quad format; the mask length is an integer from 0 to 32, inclusive. Note that the "" is required.

**ge length** - specifies that the IPv4 address matches (prefix, mask) pairs with exactly the same prefix address and mask lengths that are greater than or equal to the value of the ge command. This value can be in the range of 0 to 32, but it must be at least the value of masklen and no greater than the value of le.
le length - specifies that the IPv4 address matches (prefix, mask) pairs with exactly the same prefix address and mask lengths that are less than or equal to the value of the le command. This value can be in the range of 0 to 32, but it must be at least the value of masklen and no less than the value of ge.

Description
Prefix lists simulate a sequential lookup and return the first matched entry as the true match. The entries are ordered according to the sequential value. Sequence numbers are generated automatically unless automatic generation is disabled. If the automatic generation of sequence numbers is disabled, then the sequence number for each entry must be specified.

The optional ge and le commands can be used to specify the range of the prefix length to be matched for prefixes that are more specific than a network and netmask value. If only the ge length is specified, then the range is assumed to be from ge length to 32. If only the le length is specified, then the range is assumed to be less than the le length. An exact match is assumed when neither ge nor le is specified.

Use the negative form of this command to delete all entries or specific entries in a prefix list. One way to remove a specific entry from a prefix list is to specify all parameters that were specified when the entry was created. Another way is to specify the sequence number of the entry, which is either automatically generated or explicitly configured.

Default
Prefix lists are not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
Example 1
The following example configures a prefix list "abc" that denies all prefixes in 128.0.0.0/8 with a prefix length of 24.

    (config)# ip prefix-list abc deny 128.0.0.0/8 ge 24 le 24

Example 2
The following example configures a prefix list "abc" with three list entries. The first and second entries permit all routes matching 10.0.0.0 with prefix length equal to 8 (first entry) except 10.1.1.1 (second entry). The third entry permits all other IPv4 routes. An implicit deny is assumed for all other routes.

    (config)# ip prefix-list abc permit 10.0.0.0/8 ge 8 le 8
    (config)# ip prefix-list abc deny 10.1.1.1/32 ge 32 le 32
    (config)# ip prefix-list abc permit 0.0.0.0/0 le 32

ip prefix-list sequence-number

Name
ip prefix-list sequence-number - specifies whether Advanced Routing Suite should use automatic sequence numbering when configuring prefix lists

Syntax

    ip prefix-list sequence-number
    no ip prefix-list sequence-number

Mode
Global Configuration
Prefix Lists and Prefix Trees

Parameters
none

Description
Sequence values are generated in increments of 5, with the first sequence value generated being 5, then 10, then 15, and so on. If a sequence value is specified for an entry, and then not specified for subsequent entries, the assigned (generated) sequence values are then incremented in units of 5 based on the highest sequence number present in the list.

Note: Sequence numbers are generated automatically in increments of 5 unless automatic generation is turned off. If automatic generation is disabled, then the sequence number must be specified. Otherwise, specifying this command is optional. Use the negative form of the sequence command, no ip prefix-list sequence-number, to turn off automatic sequence number generation.

Default
Sequence numbers are generated automatically by Advanced Routing Suite. Therefore, if ip prefix-list sequence-number is not specified, it is the same as if the user had specified the following:

(config)# ip prefix-list sequence-number

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures four entries in prefix list "abc." The first entry explicitly configures sequence number 3. The two subsequent entries will, therefore, have automatically generated sequence numbers of 8 and 13 respectively. Sequence numbering is then turned off, and the fourth entry is configured with a sequence number of 20. The show ip prefix-list query that follows displays the prefix list entries with their sequence numbers.

(config)# ip prefix-list abc seq 3 permit 10.11.0.0/16 ge 20 le 20
(config)# ip prefix-list abc deny 10.10.0.0/16 ge 24 le 28
(config)# ip prefix-list abc permit 12.12.0.0 16
(config)# no ip prefix-list sequence-number
(config)# ip prefix-list abc seq 20 permit 192.168.0.0/16
(config)# exit
# show ip prefix-list abc
ip prefix-list abc: 4 entries
seq 3 permit 10.11.0.0/16 ge 20 le 20
seq 8 deny 10.10.0.0/16 ge 24 le 28
seq 13 permit 12.12.0.0/16
seq 20 permit 192.168.0.0/16
#

ip prefix-tree

Name
ip prefix-tree - configures a prefix tree

Syntax

    ip prefix-tree tree_id [deny | permit]
    [network/masklen] [ge length]? [le length]?
no ip prefix-tree list_id [deny | permit]
(network/masklen) [ge length]? [le length]?

Mode
Global Configuration

Parameters

- **tree_id** – a string of characters
- **deny | permit** – denies or permits the specified source address
- **network/masklen** – specify a valid IPv4 address with a mask length. The IPv4 address is specified in dotted-quad format; the mask length is an integer from 0 to 32, inclusive. Note that the "/" is required.
- **ge length** – specifies that the IPv4 address matches (prefix, mask) pairs with exactly the same prefix address and mask lengths that are greater than or equal to the value of the ge command. This value can be in the range of 0 to 32, but it must be at least the value of masklen and no greater than the value of le.
- **le length** – specifies that the IPv4 address matches (prefix, mask) pairs with exactly the same prefix address and mask lengths that are less than or equal to the value of the le command. This value can be in the range of 0 to 32, but it must be at least the value of masklen and no less than the value of ge.

Description

Prefix Trees return the longest match as the true match whereas Prefix Lists simulate a sequential lookup and return the first matched entry as the true match.

The optional **ge** and **le** commands can be used to specify the range of the prefix length to be matched for prefixes that are more specific than a network and netmask value. If only the **ge** length is specified, then the range is assumed to be from **ge** length to 32. If only the **le** length is specified, then the range is assumed to be less than the **le** length.

Use the negative form of this command to delete all entries or specific entries in a prefix tree. You can remove a specific entry from a prefix tree by specifying all parameters that were specified when the entry was created.

Default

Prefix trees are not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

**Example 1**
The following example configures a prefix tree "abc" that denies all prefixes in 128.0.0.0/8 with a prefix length of 24.

```plaintext
(config)# ip prefix-tree abc deny 128.0.0.0/8 ge 24 le 24
```

**Example 2**
The following example configures a prefix tree "abc" with three list entries. The first and second entries permit all routes matching 10.0.0.0 with prefix length equal to 8 (first entry) except 10.1.1.1 (second entry). The third entry permits all other IPv4 routes. An implicit deny is assumed for all other routes.

```plaintext
(config)# ip prefix-tree abc permit 10.0.0.0/8 ge 8 le8
(config)# ip prefix-tree abc deny 10.1.1.1/32 ge 32 le 32
(config)# ip prefix-tree abc permit 0.0.0.0/0 le 32
```
show ip prefix-list

Name
show ip prefix-list - displays information about all or specific prefix lists

Syntax
show ip prefix-list [ detail | summary ]? [ list_id ]?

Mode
Privileged Execution

Parameters
detail | summary - optionally specify whether you want the query to respond with a more verbose format. Summary information is returned by default.
list_id - a string of characters or an integer that represents the ID of a configured prefix list. Specifying this is optional. Advanced Routing Suite will return information for all configured IPv4 prefix lists if a list_id is not specified.

Description
The show ip prefix-list query displays information about all or specific prefix lists.

This query has two forms. If it is issued without arguments, then information about all configured IPv4 prefix lists is returned. Alternatively, the query can be issued naming a specific prefix list. In this case, the reply will contain information pertaining only to the referenced prefix list.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following query is a request for summary information about a prefix list named "pfxtest1".

    # show ip prefix-list pfxtest1
    ip prefix-list pfxtest1: 3 entries
    seq 3 deny 10.10.0.0/16 ge 20 le 20
    seq 8 permit 10.10.0.0/16 ge 24 le 28
    seq 13 permit 12.12.0.0/16

Example 2
The following query is a request for summary information about all configured IPv4 prefix lists.

    # show ip prefix-list
    ip prefix-list pfxtest1: 3 entries
    seq 3 deny 10.10.0.0/16 ge 20 le 20
    seq 8 permit 10.10.0.0/16 ge 24 le 28
    seq 13 permit 12.12.0.0/16
    ip prefix-list pfxtest2: 4 entries
    seq 3 permit 10.2.2.2/24 ge 24 le 24
    seq 8 deny 10.1.1.1/32 ge 32 le 32
    seq 10 permit 10.0.0.0/8 ge 8 le 8
    seq 11 permit 0.0.0.0/8 le 8

Example 3
The following is a request for detail information about a prefix list named "pfxtest2".
show ip prefix-tree

Name
show ip prefix-tree - displays information about all or specific prefix trees

Syntax
    show ip prefix-tree [ detail | summary ]? [ tree_id ]?

Mode
Privileged Execution

Parameters
    detail | summary - optionally specify whether you want the query to respond with a more verbose format. Summary information is returned by default.
    tree_id - a string of characters or an integer that represents the ID of a configured prefix tree. Specifying this is optional. Advanced Routing Suite will return information for all configured IPv4 prefix trees if a tree_id is not specified.

Description
The show ip prefix-tree query displays information about all or specific prefix trees.
This query has two forms. If it is issued without arguments, then summary information about all configured IPv4 prefix trees is returned. Alternatively, the query can be issued naming a specific prefix tree. In this case, the reply will contain information pertaining only to the referenced prefix tree.

Command History
NGC 2.2 - This command was introduced.

Examples

Example 1
The following query is a request for summary information about a prefix tree named "pfxtest1".
    # show ip prefix-tree pfxtest1
    ip prefix-tree pfxtest1: 3 entries
deny 10.10.0.0 0.0.255.255 ge 20 le 20
permit 10.10.0.0 0.0.255.255 ge 24 le 28
permit 12.12.0.0 0.0.255.255

Example 2
The following query is a request for summary information about all configured IPv4 prefix trees.
    # show ip prefix-tree
    ip prefix-tree pfxtest1: 3 entries
deny 10.10.0.0 0.0.255.255 ge 20 le 20
permit 10.10.0.0 0.0.255.255 ge 24 le 28
permit 12.12.0.0 0.0.255.255
Example 3

The following is a request for detail information about a prefix tree named "pfxtest2".

```
# show ip prefix-tree detail pfxtest2

ip prefix-tree pfxtest2:
  count: 4, sequences 3 - 11
  permit 10.2.2.2 0.255.255.255 ge 24 le 24
  deny 10.1.1.1 255.255.255.255 ge 32 le 32
  permit 10.0.0.0 0.0.0.255 ge 8 le 8
  permit 0.0.0.0 0.0.0.255 le 8
```
Chapter 24

Route Aggregation and Generation

In This Chapter

Route Aggregation and Generation Overview
aggregate-address
router aggregate

Route Aggregation and Generation Overview

Route aggregation is a method of generating a more general summary route, given the presence of a more specific route. It is used, for example, at an autonomous system border to generate a route to a network to be advertised via BGP, given the presence of one or more subnets of that network learned via RIP. No aggregation is performed unless explicitly requested in an aggregate command.

Route aggregation is also used by regional and national networks to reduce the amount of routing information passed. With careful allocation of network addresses to clients, regional networks can announce one route to regional networks instead of hundreds.

Aggregate routes are not actually used for packet forwarding by the originator of the aggregate route, only by the receiver (if it wishes). A router is supposed to respond with an ICMP network unreachable message if the router receives a packet that does not match one of the component routes that led to the generation of an aggregate route. This message is to prevent packets for unknown component routes from following a default route into another network, where they would be forwarded back to the border router, and around and around again and again, until their TTL expired. Sending an unreachable message for a missing piece of an aggregate is only possible on systems with support for reject routes.

A difference between route aggregation and generation is the generation of a route based on the existence of certain conditions. This is sometimes known as the "route of last resort." This route inherits the next hops and AS path from the contributor specified with the lowest (most favorable) preference. The most common usage for this is to generate a default based on the presence of a route from a peer on a neighboring backbone.

aggregate-address

Named

aggregate-address - allows the aggregation of specific routes into one

Syntax

aggregate-address [(address (mask | masklen)?) | prefix ] protocol [aggregate | all | bgp | direct | kernel | ospf | rip | ripng | static]
[match-map map_name]? (rib [unicast | multicast | unicast-multicast])? [preference pref]? generate?
noinstall? blackhole? bgp? brief?

no aggregate-address [(address (mask | masklen)?) | prefix ] protocol [aggregate | all | bgp | direct | isis | kernel | ospf | rip | ripng | static]
[match-map map_name]? (rib [unicast | multicast |

Mode
Aggregate Router Configuration Mode

Parameters
address (mask | masklen) - the IPv4 address to be aggregated or generated. Specify either an address or a prefix. The address can be accompanied by an address mask or masklen. The address mask is specified in dotted-quad format for IPv4 addresses. The masklen specifies the number of contiguous bits at the beginning of a mask.

prefix - specifies a valid IPv4 prefix

protocol - specify the protocol whose rules this aggregate will use to determine whether to include each route. For IPv4 routes, valid values include the following: aggregate, all, bgp, direct, kernel, ospf, rip, and static.

match-map map_name - optionally specify a route map that will be used to generate this route

rib [ unicast | multicast | unicast-multicast ] - specifies whether the aggregate is restricted to the unicast RIB, the multicast RIB, or both. The default is to restrict the aggregate to both the unicast and multicast RIB.

brief - specifies that the AS path should be truncated to the longest common leading AS path. This is option is only useful if the configured protocol is BGP.

bgp - specifies that this aggregate will use BGP rules to determine whether to include each route. This is a valid option only if the configured protocol is BGP.

preference pref - specifies the aggregation preference value to be assigned to the contributor route. This can be an integer between 1 and 255, inclusive. Note: If two different aggregate commands are specified for the same prefix, then the preference specified most recently will be used by Advanced Routing Suite, and that preference value will display in the show running-config query for each configured aggregate route.

noinstall - specifies whether to install the generate address if any contributor route is active. The default behavior is to install the route.

generate - generates a default route. Route generation is a method of generating a more general route, given the presence of a specific route. generate differs from aggregate only in terms of the route installed in the kernel.

blackhole - specifies whether to blackhole the route; thus enabling the router to refuse to route various prefixes

Description
Route aggregation is a method of generating a more general summary route, given the presence of a more specific route. generate differs from aggregate only in terms of the route installed in the kernel.

Note: Using the generate keyword in the no form restores the default for the generate route but has no effect on the aggregate route. To restore the default for the aggregate route, use the no form of the command without the generate keyword. This will have a cascading effect on restoring the default generated route.

Default
No aggregate/generate routes are configured by default.

Command History
NGC 2.2 - This command was introduced.
Examples

Example 1
In the following example, an aggregate route to 10/8 is generated based on routes formed via BGP that match route map "rm1".

```
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
   protocol bgp match-map rm1
(config-aggregate)# exit
(config)#
```

Example 2
The following example removes the configuration in Example 1 and restores the defaults.

```
(config)# router aggregate
(config-aggregate)# no aggregate-address 10.0.0.0 255.0.0.0
   protocol bgp
(config-aggregate)# exit
(config)#
```

Example 3
In the following example, a generate route to 10/8 is created based on routes formed via BGP that match route map "rm1".

```
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
   protocol bgp match-map rm1 generate
(config-aggregate)# exit
(config)#
```

Example 4
The following example removes the configuration in Example 3 and restores the defaults.

```
(config)# router aggregate
(config-aggregate)# no aggregate-address 10.0.0.0 255.0.0.0
   protocol bgp generate
(config-aggregate)# exit
(config)#
```

Example 5
The following example creates an aggregate route that includes all OSPF routes matching route map "rm1" and that are more specific than 10/8. It also restricts it to the unicast RIB.

```
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
   protocol ospf match-map rm1 rib unicast
(config-aggregate)# exit
(config)#
```

Example 6
The following example creates a generate route from all OSPF routes that match route map "rm1" and that are more specific than 10/0. It restricts the generate route to the unicast RIB. The routes, however, will not be installed if the contributor route is not active.
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0 protocol ospf
match-map rm1 rib unicast generate noinstall
(config-aggregate)# exit
(config)#

Example 7
The following example restores the defaults in Example 6.
(config)# router aggregate
(config-aggregate)# no aggregate-address 10.0.0.0 255.0.0.0
protocol ospf rib unicast generate
(config-aggregate)# exit
(config)#

Example 8
The following example creates an aggregate route that includes all ISIS routes matching route map "rm1" and that are more specific than 10/8. In addition, it sets the preference for these routes to 10.
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
protocol isis
match-map rm1 preference 10
(config-aggregate)# exit
(config)#

Example 9
The following example creates an aggregate route that includes all RIP routes matching route map "rm1" and that are more specific than 10/8. It also sets the preference for the aggregate route to 10.
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
protocol rip
match-map rm1 preference 10
(config-aggregate)# exit
(config)#

Example 10
The following example creates a generate route from all RIP routes matching route map "rm1" and that are more specific than 10/8. It also sets the preference for these routes to 10. The routes, however, will not be installed if the contributor route is not active.
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
protocol rip
match-map rm1 preference 10 generate noinstall
(config-aggregate)# exit
(config)#

Example 11
The following example creates an aggregate route that includes all static routes matching route map "rm1" and that are more specific than 10/8. It also sets the preference for the aggregate route to 10.
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
protocol static
match-map rm1 preference 10
(config-aggregate)# exit
Example 12
The following example creates a generate route from all static routes that match route map "rm1" and that are more specific than 10/8. It also sets the preference for these routes to 10.

```
(config)# router aggregate
(config-aggregate)# aggregate-address 10.0.0.0 255.0.0.0
protocol static match-map rm1 preference 10 generate
(config-aggregate)# exit
(config)#
```

router aggregate

Name
routing aggregate - enters the user into Aggregate Router Configuration mode

Syntax
```
router aggregate
```

Mode
Global Configuration

Parameters
none

Description
Use the router aggregate command to enter Aggregate Router Configuration mode. This mode allows the aggregation of specific routes into a single route.

Default
This command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example shows how to enter Aggregate Router Configuration mode.
```
(config)# router aggregate
(config-aggregate)#
```
Chapter 25

Route Flap Damping

In This Chapter

Route Flap Damping Overview .......................................................... 449
dampen-flap  ............................................................................. 449
keep-history .............................................................................. 450
max-flap ................................................................................. 451
reach-decay ............................................................................... 451
reach-tick .................................................................................. 452
reuse-below .............................................................................. 453
suppress-above ................................................................. 454
unreach-decay ............................................................................. 454

Route Flap Damping Overview

Weighted route dampening treats routes that are being announced and withdrawn (flapping) at a rapid rate as unreachable. If a route flaps at a low rate, it should not be suppressed at all, or suppressed for only a brief period of time. With weighted route dampening, the suppression of a route or routes occurs in a manner that adapts to the frequency and duration that a particular route appears to be flapping. The more a route flaps during a period of time, the longer it will be suppressed. The adaptive characteristics of weighted route dampening are controlled by a few tags.

Note: Route flap damping currently only applies to BGP.

dampen-flap

Name
dampen-flap - enters the user into Dampen Flap Configuration mode

Syntax
dampen-flap name
no dampen-flap name

Mode
Global Configuration

Parameters
name - a string of characters uniquely identifying a dampen-flap configuration

Description
The dampen-flap command enters the user into Dampen Flap Configuration mode, where route-flap-damping parameters can be specified. The negative form of this command, no dampen-flap, removes the configured damping.
Default
This command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures route flap damping with an ID of rfd1.

```
(config)# dampen-flap rfd1
(config-dampen-flap)#
```

keep-history

Name
keep-history - specifies the time in seconds for which any history of a route's instability is maintained

Syntax
```
keep-history time-seconds
no keep-history time-seconds?
```

Mode
Dampen Flap Configuration

Parameters
time-seconds - specifies a time in seconds from 1 to 604800, inclusive

Description
The keep-history command specifies time in seconds for which any history of a route's instability is maintained. The negative form of this command, no keep-history, removes the configured value and returns this to its default value of 1800 seconds. **Note:** Specifying a value for time-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If keep-history is not specified, it is the same as if the user had specified the following:

```
(config-dampen-flap)# keep-history 1800
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the keep-history value to be 2000 seconds.

```
(config)# dampen-flap rfd1
(config-dampen-flap)# keep-history 2000
```
**max-flap**

**Name**

`max-flap` - specifies the maximum value of a route's instability history

**Syntax**

```
max-flap float
no max-flap float?
```

**Mode**

Dampen Flap Configuration

**Parameters**

`float` - an integer or floating point number between 2.1 and 604800, inclusive

**Description**

Use the `max-flap` command to specify the maximum value of a route's instability history. The instability metric is incremented by 1 each time a route becomes unreachable. The `max-flap` value, which must be greater than the `suppress-above` value, determines the longest time that a route can be suppressed.

The negative form of this command, `no max-flap`, removes the configured value and returns this to its default value of 16.0. **Note:** Specifying a value for `float` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `max-flap` is not specified, it is the same as if the user had specified the following:

```
(config-dampen-flap)# max-flap 16.0
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the maximum value of the route's instability history to be 18.

```
(config)# dampen-flap rfd1
(config-dampen-flap)# max-flap 18
```

**See Also**

"suppress-above" ("suppress-above" on page 454)

**reach-decay**

**Name**

`reach-decay` - specifies the time in seconds after which a reachable route's instability history decays to half of its current value

**Syntax**

```
reach-decay time-seconds
no reach-decay time-seconds?
```
Mode
Dampen Flap Configuration

Parameters

$time$-seconds
- specifies a time in seconds from 1 to 604800, inclusive

Description
Use the reach-decay command to specify a time in seconds, after which a reachable route’s instability history decays to half of its current value. The negative form of this command, no reach-decay, removes the configured value and returns this to its default value of 300 seconds. Note: Specifying a value for $time$-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If reach-decay is not specified, it is the same as if the user had specified the following:

```
(config-dampen-flap)# reach-decay 300
```

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the reach-decay value to be 600 seconds.

```
(config)# dampen-flap rfd1
(config-dampen-flap)# reach-decay 600
```

reach-tick

Name
reach-tick
- specifies the granularity of instability history in seconds

Syntax

```
reach-tick time-seCONDS
no reach-tick time-seCONDS?
```

Mode
Dampen Flap Configuration

Parameters

$time$-seconds
- an integer between 1 and 604800, inclusive, used to specify the time in seconds

Description
The reach-tick command specifies the granularity of instability history in seconds. The negative form of this command, no reach-tick, removes the configured value and returns this to its default value of 1 second. Note: Specifying a value for $time$-seconds in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If reach-tick is not specified, it is the same as if the user had specified the following:

```
(config-dampen-flap)# reach-tick 1
```
**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the reach-tick value to be 5 seconds
```
(config)# dampen-flap rfd1
(config-dampen-flap)# reach-tick 5
```

### reuse-below

**Name**
reuse-below - specifies the value of a route’s instability, below which a suppressed route is reused

**Syntax**
```
reuse-below float
no reuse-below float?
```

**Mode**
Dampen Flap Configuration

**Parameters**

*float* - a floating point number whose value is less than or equal to the `suppress-above` value, and in the range of 1.0 to 604800, inclusive

**Description**
The `reuse-below` command specifies the value of a route’s instability metric, below which a suppressed route is reused. The value for this command must be less or equal to the `suppress-above` value.

The negative form of this command, `no reuse-below`, removes the configured value and returns this to its default value of 1.0. **Note:** Specifying a value for `float` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**
If `reuse-below` is not specified, it is the same as if the user had specified the following:
```
(config-dampen-flap)# reuse-below 1.0
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures the `reuse-below` value to be 6
```
(config)# dampen-flap rfd1
(config-dampen-flap)# reuse-below 6
```

**See Also**
"supress-above" ("suppress-above" on page 454)
suppress-above

Name
suppress-above - specifies the value of a route's instability, above which the route is suppressed

Syntax
suppress-above float
no suppress-above float?

Mode
Dampen Flap Configuration

Parameters
float - a floating point number whose value is greater than or equal to the reuse-above value, and in the range of 2.0 to 604800, inclusive

Description
The suppress-above command specifies the value of a route's instability metric, above which the route is suppressed. The negative form of this command, no suppress-above, removes the configured value and returns this to its default value of 2.0. Note: Specifying a value for float in the no form has no effect on the configuration. Thus, it is displayed above as optional.

Default
If suppress-above is not specified, it is the same as if the user had specified the following:
(config-dampen-flap)# suppress-above 2.0

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures the suppress-above value to be 8.
(config)# dampen-flap rfd1
(config-dampen-flap)# suppress-above 8

See Also
"reuse-below" ("reuse-below" on page 453)

unreach-decay

Name
unreach-decay - specifies the time in seconds after which an unreachable route's instability history decays to half its current value

Syntax
unreach-decay time-seconds
no unreach-decay time-seconds?

Mode
Dampen Flap Configuration
**Parameters**

`time-seconds` - specifies a time in seconds from 1 to 604800, inclusive

**Description**

Use the `unreach-decay` command to specify a time in seconds, after which a unreachable route’s instability history decays to half of its current value. The negative form of this command, `no unreach-decay`, removes the configured value and returns this to its default value of 900 seconds. **Note:** Specifying a value for `time-seconds` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

**Default**

If `unreach-decay` is not specified, it is the same as if the user had specified the following:

```
(config-dampen-flap)# unreach-decay 900
```

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example configures the `unreach-decay` value to be 600 seconds.

```
(config)# dampen-flap rfd1
(config-dampen-flap)# unreach-decay 600
```
Chapter 26

Route Maps

In This Chapter

Route Maps Overview 457
match aggregate-contributors 457
match as 458
match as-path 458
match as-path-list 459
match community 460
match community-set 461
match distance 462
match extended-community-set 463
match instance 464
match interface 465
match ip address access-list 466
match ip address prefix-list 466
match ip address prefix-tree 467
match ip gateway 468
match ip next-hop 469
match ip route-source prefix-tree 470
match localpref 471
match med 471
match metric 472
match metric-type 473
match protocol 473
match ribs 474
match tag 475
route-map 476
set as-path prepend 477
set community-set 477
set dampen-flap 479
set ip next-hop 479
set local-preference 480
set med 481
set metric 482
set metric-type 482
set origin 483
set preference 484
set propagate 485
set ribs 486
set tag 486
Route Maps Overview

Route maps can be used to filter learned routes or to redistribute routes between protocols. Route map commands specify match and set criteria associated with the route map.

match aggregate-contributors

Name
match aggregate-contributors - specifies that in order for a route to match, it must be a contributor to an aggregate route

Syntax

match aggregate-contributors [invert]?
no match aggregate-contributors [invert]?

Mode
Route Map Configuration

Parameters
invert - optionally specify to reverse the result of a match

Description
Use the match aggregate-contributors command to specifies that, in order for a route to match, it must be a contributor to an aggregate route.

Default
This command is configured by default. Therefore, if match aggregate-contributors is not specified, it is the same as if the user had specified the following:

(config-route-map)# match aggregate-contributors

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route being matched is an aggregate contributor and has an AS_PATH matching the AS path list labeled "list1", then the following set actions will be executed:

- Communities specified in community "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2 will be deleted from the matching route.
- The metric for the matching routes will be set to 1.

(config)# route-map abc
(config-route-map)# match aggregate-contributors
(config-route-map)# match as-path-list list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 1
(config-route-map)# exit
(config)#
match as

Name
match as - filters BGP routes learned from peers within a particular autonomous system (AS)

Syntax
match as as_number [invert]?
no match as as_number [invert]?

Mode
Route Map Configuration

Parameters
as_number - an integer between 1 and 65535, inclusive, specifying the number of an AS
invert - optionally specify to reverse the result of a match

Description
Use the match as command to filter BGP routes learned from peers within a particular autonomous system.

Note: This command applies only to BGP routes. It will be ignored for any other protocol.

Default
AS matching policy is not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example configures route map "matchbgp" and specifies to match BGP routes learned from peers within AS 65535.

```
(config)# route map matchbgp
(config-route-map)# match as 65535
(config-route-map)# exit
(config)#
```

match as-path

Name
match as-path - specifies the name of the AS Path Regular Expression to be matched

Syntax
match as-path name [invert]?
no match as-path name [invert]?

Mode
Router Map Configuration
Parameters

name - the name of the AS Path Regular Expression to be matched
invert - optionally specify to reverse the result of a match

Description

Use the match as-path command to specify a pre-configured AS Path Regular Expression that you want to be matched in this route map. If the referenced AS Path Regular Expression has not yet been configured, then the result is that the match fails.

Note: This command applies only to BGP routes. It will be ignored for all other protocols.

Default

No AS Path matches are configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if the route being evaluated against route map "abc" matches interface "fxp1" and the AS path regular expression "aspath1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match as-path aspath1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```

match as-path-list

Name

match as-path-list - specifies a pre-configured AS Path list to be matched

Syntax

```
match as-path-list name [invert]?
no match as-path-list name [invert]?
```

Mode

Route Map Configuration

Parameters

name - the name of a pre-configured AS Path list
invert - optionally specify to reverse the result of a match
**Description**

Use the `match as-path-list` command to specify a pre-configured AS Path list that you want to be matched in this route map. If the referenced AS Path list has not yet been configured, then the result is that the match fails.

**Note:** This command applies only to BGP routes. It will be ignored for all other protocols.

**Default**

No AS Path lists are matched by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route being evaluated against route map "abc" matches interface "fxp1" and the AS path list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match as-path-list list1
(config-route-map)# set community com-set-1 additive
(config-route-map)# set community com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```

**match community**

**Name**

`match community` - specifies a BGP community list to be matched in a route map

**Syntax**

```
match community name [invert]?
no match community name [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `name` - the name of a BGP community list
- `invert` - optionally specify to reverse the result of a match

**Description**

Use the `match community` command to specify a BGP community list that you want to be matched in this route map. If the referenced community list has not yet been configured, then the result is that the match fails.
**Note:** This command applies only to BGP routes. It will be ignored for all other protocols.

**Default**
No BGP community lists are matched by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

**Example 1**
In the following example, if the route being evaluated against route map "abc" matches the BGP Community List "commlist1" and the AS path list labeled "aspathlist1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match community commlist1
(config-route-map)# match as-path-list aspathlist1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```

**Example 2**
The following example configures route map "abc". If a route being evaluated against this route map does not match community list "comm-list-1", then communities specified in the community set "comm-set-1" will be deleted from the route, and the route will be considered to have matched.

```
(config)# route-map abc
(config-route-map)# match community comm-list-1 invert
(config-route-map)# set community-set comm-set-1 delete
(config-route-map)# exit
(config)#
```

**match community-set**

**Name**

match community-set - specifies a set of BGP communities to be matched

**Syntax**

```
match community-set name [exact]? [invert]?
no match community-set name [exact]? [invert]?
```

**Mode**
Route Map Configuration
Parameters

- **name** - the name of the community set
- **exact** - optionally specify that a community set must be matched exactly. Ordinarily, supersets are considered to match a community set.
- **invert** - optionally specify to reverse the result of a match

Description

Use the `match community-set` command to specify a set of BGP communities that you want to be matched in this route map. If the referenced community set has not yet been configured, then the result is that the match fails.

**Note:** This command applies only to BGP routes. It will be ignored for all other protocols.

Default

No community sets are matched by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if the route being evaluated against route map "abc" matches interface "fxf1" and the BGP Community labeled "bgpcomm1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in the community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxf1
(config-route-map)# match community-set bgpcomm1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```

**match distance**

**Name**

- `match distance` - specifies to match a route's distance (or preference) in the route map

**Syntax**

```
match distance dist [invert]?
no match distance dist [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- **dist** - an integer from 0 to 255, inclusive
invert - optionally specify to reverse the result of a match

**Description**
This command specifies to match a route's preference in this route map.

**Default**
Distances (or preferences) are not matched by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
The following example configures a route map, xyz, that specifies to match routes with a distance of 10.

```
(config)# route-map xyz
(config-route-map)# match distance 10
(config-route-map)# exit
(config)#
```

**match extended-community-set**

**Name**
match community-set - specifies a set of BGP extended communities to be matched

**Syntax**
match extended-community-set name [exact]? [invert]?
no match extended-community-set name [exact]? [invert]?

**Mode**
Route Map Configuration

**Parameters**

name - the name of the community set

exact - optionally specify that a community set must be matched exactly. Ordinarily, supersets are considered to match a community set.

invert - optionally specify to reverse the result of a match

**Description**
Use the match extended-community-set command to specify a set of BGP extended communities that you want to be matched in this route map. If the referenced extended community set has not yet been configured, then the result is that the match fails.

**Note:** This command applies only to BGP routes. It will be ignored for all other protocols.

**Default**
No extended community sets are matched by default.

**Command History**
NGC 2.2 - This command was introduced.
Examples
In the following example, if the route being evaluated against route map "abc" matches interface "fxp1" and the BGP Extended Community labeled "extcomm1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in the community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match extended-community-set extcomm1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```

**match instance**

**Name**

`match instance` - specifies an OSPF instance to be matched by a route map

**Syntax**

```
match instance instance-id [invert]?
no match instance instance-id [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `instance-id` - an integer from 1 to 65535, inclusive, specifying an OSPF instance
- `invert` - optionally specify to reverse the result of a match

**Description**

Use the `match instance` command to specify an OSPF instance to be matched. **Note:** This command applies only to OSPF. It will be ignored for all other protocols.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route matches instance 1, then the route will only be exported to level 1.

```
(config)# route-map abc
(config-route-map)# match instance 1
(config-route-map)# set level 1
```
**match interface**

**Name**
match interface - specifies an interface or set of interfaces to be matched

**Syntax**

```
match interface [interface_name | all | ip_address] [invert]? {1,n}
no match interface [interface_name | all | ip_address] [invert]? {1,n}
```

**Mode**
Route Map Configuration

**Parameters**

- **interface_name** - specify the physical interface name
- **all** - specify to include match all interfaces
- **ip_address** - specify a valid IPv4 logical interface address
- **invert** - optionally specify to reverse the result of a match
- **{1,n}** - You can use this command an unlimited number of times to specify an unlimited number of interfaces

**Description**
Use `match interface` to specify an interface or set of interfaces that you want to be matched in this route map.

**Default**
This option is not explicitly configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
In the following example, if the route being evaluated against route map "abc" matches interface "fxp1" and the AS path list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The metric for the matching route will be set to 50.

```
(config)# route-map abc
(config-route-map)# match interface fxp1
(config-route-map)# match as-path-list list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 50
(config-route-map)# exit
(config)#
```
match ip address access-list

Name
match ip address access-list - specifies an access list to be matched in this route map

Syntax
match ip address access-list name [invert]?
no match ip address access-list name [invert]?

Mode
Route Map Configuration

Parameters
name - the name of the access list to be matched
invert - optionally specify to reverse the result of a match

Description
Use the match ip address access-list command to specify an access list that you want to be matched in this route map. If the referenced access list has not yet been configured, then the result is that the match fails.

Default
Access list matches are not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route matches the prefix list labeled, "list1" and the access list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The MED for routes will be set to 1.

```
(config)# route-map abc
(config-route-map)# match ip address prefix-list list1
(config-route-map)# match ip address access-list list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 1
(config-route-map)# exit
(config)#
```

match ip address prefix-list

Name
match ip address prefix-list - specifies a prefix list to be matched in this route map
Syntax

```
match ip address prefix-list name [invert]?
no match ip address prefix-list name [invert]?
```

Mode
Route Map Configuration

Parameters

- `name` - the name of the prefix list to be matched
- `invert` - optionally specify to reverse the result of a match

Description

Use the `match ip address prefix-list` command to add a prefix list that you want to be matched in this route map. If the referenced prefix list has not yet been configured, then the result is that the match fails.

Default
Prefix list matches are not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if the route matches the prefix list labeled "list1" and the access list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The MED for the matching route will be set to 1.

```
(config)# route-map abc
(config-route-map)# match ip address prefix-list list1
(config-route-map)# match ip address access-list list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 1
(config-route-map)# exit
(config)#
```

**match ip address prefix-tree**

Name

- `match ip address prefix-tree` - specifies a prefix tree to be matched in this route map

Syntax

```
match ip address prefix-tree name [invert]?
no match ip address prefix-tree name [invert]?
```

Mode
Route Map Configuration
Parameters

*name* - the name of the prefix tree to be matched

*invert* - optionally specify to reverse the result of a match

Description

Use the `match ip address prefix-tree` command to add a prefix tree that you want to be matched in this route map. If the referenced prefix tree has not yet been configured, then the result is that the match fails.

Default

Prefix tree matches are not configured by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if the route matches the prefix tree labeled, "tree1" and the access list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The MED for the matching route will be set to 1.

```bash
(config)# route-map abc
(config-route-map)# match ip address prefix-tree tree1
(config-route-map)# match ip address access-list list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 1
(config-route-map)# exit
(config)#
```

**match ip gateway**

Name

`match ip gateway` - specifies a gateway to be matched

Syntax

```
match ip gateway ipv4_address [invert]?
no match ip gateway ipv4_address [invert]?
```

Mode

Route Map Configuration

Parameters

*ipv4_address* - specifies a valid IPv4 logical address for the gateway

*invert* - optionally specify to reverse the result of a match
**Description**

Use the `match ip gateway` command to match IPv4 gateways and to specify those gateway addresses.

**Default**

Gateway values are not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example specifies to match gateway 1.2.3.4 in route map abc.

```
(config)# route-map abc
(config-route-map)# match ip gateway 1.2.3.4
(config-route-map)# exit
(config)#
```

---

**match ip next-hop**

**Name**

`match ip next-hop` - specifies to match a nexthop address in a route map

**Syntax**

```
match ip [bgp]? next-hop [ipv4_address | all] [invert]?
no match ip [bgp]? next-hop [ipv4_address | all] [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `bgp` - optionally specify to match a BGP nexthop address in the route map
- `ipv4_address` - specifies a valid IPv4 logical address for the nexthop
- `all` - specifies that all nexthops of a route must match
- `invert` - optionally specify to reverse the result of a match

**Description**

Use the `match ip next-hop` command to specify the IPv4 nexthop address to be matched.

**Default**

Nexthop values are not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following example specifies to match nexthop 1.2.3.4 in route map abc.

```
(config)# route-map abc
(config-route-map)# match ip next-hop 1.2.3.4
```
match ip route-source prefix-tree

Name
match ip route-source prefix-tree — specifies the name of a prefix tree to be matched

Syntax
match ip route-source prefix-tree name [invert]?
no match ip route-source prefix-tree name [invert]?

Mode
Route Map Configuration

Parameters
name — the name of the prefix-tree to be matched
invert — optionally specify to reverse the result of a match

Description
Use match ip route-source prefix-tree to specify prefix trees that you want to be matched in the associated route map. If the referenced prefix tree has not yet been configured, then the result is that the match fails.

Default
Prefix tree matches are not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route matches the community list labeled, "list1" and the prefix tree labeled "tree1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The MED for the matching route will be set to 1.

```
(config)# route-map abc
(config-route-map)# match ip route-source prefix-tree tree1
(config-route-map)# match community list1
(config-route-map)# set community-set com-set-1 additive
(config-route-map)# set community-set com-set-2 delete
(config-route-map)# set metric 1
(config-route-map)# exit
(config)#
```
match localpref

**Name**

`match localpref` - specifies a BGP Local_Pref value to be matched

**Syntax**

```
match localpref value [invert]?
no match localpref value [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `value` - an integer from 0 to 4,294,967,295, inclusive
- `invert` - optionally specify to reverse the result of a match

**Description**

Use the `match localpref` command to filter BGP routes with a specific Local_Pref value.

**Note:** This command applies only to BGP routes. It will be ignored for any other protocol.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route has a BGP Local Preference value of 20, then the MED will be set to 1.

```
(config)# route-map abc
(config-route-map)# match localpref 20
(config-route-map)# set med 1
(config-route-map)# exit
```

match med

**Name**

`match med` - specifies a BGP Multi-Exit Discriminator (MED) to be matched

**Syntax**

```
match med value [invert]?
no match med value [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `value` - an integer from 0 to 4,294,967,295, inclusive
invert - optionally specify to reverse the result of a match

**Description**

Use the `match med` command to specify a BGP MED value to be matched. **Note:** This command applies only to BGP. It will be ignored for all other protocols.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route has a BGP MED value of 20, then the Local Pref value will be set to 1.

```
(config)# route-map abc
(config-route-map)# match med 20
(config-route-map)# set local-preference 1
(config-route-map)# exit
```

**match metric**

**Name**

`match metric` - specifies a route metric to be matched

**Syntax**

```
match metric value [invert]?
no match metric value [invert]?
```

**Mode**

Route Map Configuration

**Parameters**

- `value` - an integer from 0 to 4,294,967,295, inclusive
- `invert` - optionally specify to reverse the result of a match

**Description**

Use the `match metric` command to specify a route's metric value to be matched.

**Default**

This command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route has a metric value of 4, then the preference will be set to 50.

```
(config)# route-map xyz
(config-route-map)# match metric 4
```
(config-route-map)# set preference 50
(config-route-map)# exit
(config)#

**match metric-type**

**Name**
match metric-type - compares the metric types for OSPF

**Syntax**
match metric-type [type-1 | type-2] [invert]?
no match metric-type [type-1 | type-2] [invert]?

**Mode**
Route Map Configuration

**Parameters**
type-1 - specifies to match the OSPF external type 1 metric
type-2 - specifies to match the OSPF external type 2 metric
invert - optionally specify to reverse the result of a match

**Description**
Use the match metric-type command when comparing metric types for OSPF. The meanings are specific to each protocol.

**Note:** You cannot include more than one parameter for a single match metric-type command

**Default**
If match metric-type is not specified, it is the same as if the user had specified the following for OSPF: (config-route-map)# match metric-type type-2

**Command History**
NGC 2.2 - This command was introduced.

**match protocol**

**Name**
match protocol - specifies a protocol to be matched

**Syntax**
match protocol proto [invert]?
match protocol proto [invert]?

**Mode**
Route Map Configuration

**Parameters**
proto - specifies a protocol to be matched. Valid values include the following:
aggregate – Aggregate routes
all – All protocols
bgp – BGP routes
direct – Directly connected routes
kernel – Kernel routes
ospf – OSPF routes
ospf-ase – OSPF ASE routes
ospf3-ase – OSPFv3 ASE routes
rip – RIP routes
ripng – RIPng routes
static – Static routes
invert – optionally specify to reverse the result of a match

Description
Use the match protocol command to specify routes from a specific protocol to be matched.

Default
This command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route is a BGP route, then the preference will be set to 50.

```
(config)# route-map xyz
(config-route-map)# match protocol bgp
(config-route-map)# set preference 50
(config-route-map)# exit
(config)#
```

match ribs

Name
match ribs – specifies RIBs to be matched

Syntax
```
match ribs [unicast || multicast] [invert]?
no match ribs [unicast || multicast] [invert]?
```

Mode
Route Map Configuration

Parameters
unicast – specifies unicast RIB
multicast – specifies multicast RIB
invert – optionally specify to reverse the result of a match
Description

Use the `match ribs` command to limit the match to a set of either unicast and/or multicast RIBs. By default, a route map applies to all RIBs to which any contributing route applies. For example, a route map applies to the Unicast RIB if and only if any contributing route applies to the Unicast RIB.

Note: If a `set ribs` command is not specified, then no action is taken if the routes match.

Default

If `match ribs` is not specified, it is as if the user had specified the following:

```
(config)# match unicast multicast
```

Command History

NGC 2.2 - This command was introduced.

Examples

The following example causes routes in the unicast RIB to be exported into both the unicast and multicast RIBs.

```
(config)# route-map abc
(config-route-map)# match ribs unicast
(config-route-map)# set ribs multicast
(config-route-map)# set ribs unicast
```

match tag

Name

`match tag` - specifies the tag value with which routes are announced in OSPF, RIP, or RIPng

Syntax

```
match tag tag_value [invert]?
no match tag tag_value? [invert]?
```

Mode

Route Map Configuration

Parameters

- `tag_value` - an integer from 0 to 4,294,967,295, inclusive
- `invert` - optionally specify to reverse the result of a match

Description

Use the `match tag` command specifies a tag value with which routes are announced in OSPF, RIP, or RIPng. The negative form of this command, `no match tag`, removes the configured `tag_value` and returns this to its default value of 0.

Notes:

- Specifying a value for `tag_value` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- This command will be ignored for any protocol that is not OSPF, RIP, or RIPng.

Default

If `match tag` is not specified, it is the same as if the user had specified the following:
(config-route-map)# match tag 0

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if routes are announced with a tag value of 4, then when this route map is imported into a protocol, the preference for that protocol will be configured as 50.

```
(config)# route-map abc
(config-route-map)# match tag 4
(config-route-map)# set preference 50
(config-route-map)# exit
(config)#
```

---

**route-map**

**Name**

route-map – enters route-map configuration mode for a specified route map

**Syntax**

```
route-map name [permit | deny]? [seq_value]?
no route-map name [permit | deny]? [seq_value]?
```

**Mode**

Global Configuration

**Parameters**

- **name** – a string of characters that identifies this route map
- **permit | deny** – accepts or rejects routes matching this route map. If neither is specified, this option defaults to permit.
- **seq_value** – an integer indicating the position a new route map is to have in a list of previously configured route maps

**Description**

The `route-map` command specifies the name of a route map and optionally whether to permit or deny routes matching the route map. A sequence number can also be specified. Specifying this command enters you in Route Map Configuration mode, where further commands can be specified.

**Default**

Route maps are not configured by default. When one is configured, however, the default behavior is to permit the route map.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

The following configuration enters Route Map Configuration mode for route map abc.

```
(config)# route-map abc
(config-route-map)#
```
set as-path prepend

**Name**

set as-path prepend – specifies a series of AS numbers to be prepended to a route’s AS_PATH before announcement

**Syntax**

```
set as-path prepend as_number
no set as-path prepend as_number
```

**Mode**

Route Map Configuration

**Parameters**

`as_number` – an AS number or list of AS numbers specified using integers. The value range is 0 to 65535, inclusive.

**Description**

Use the `set as-path prepend` command to specify a series of AS numbers to be prepended to a route’s AS_PATH before announcement. These AS numbers are included in a lending sequence of the type appropriate for the peer they are being sent to.

**Note:** This command applies only to BGP. It will be ignored for all other protocols.

**Default**

The `set as-path prepend` command is not explicitly configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if a route matches AS Path list “alist1”, then the numbers 655 will be prepended to a route’s AS_PATH before announcement, and the origin advertised to BGP will be “incomplete”.

```
(config)# route-map xyz
(config-route-map)# match as-path-list alist1
(config-route-map)# set as-path prepend 655
(config-route-map)# set origin incomplete
(config-route-map)# exit
(config)#
```

set community-set

**Name**

set community-set – specifies a community set that should be added to a route, removed from a route, or overwritten

**Syntax**

```
set community-set name [extended]? [delete | additive? | overwrite]
```
Route Maps

no set community-set name [extended]? [delete | additive | overwrite]?

Mode
Route Map Configuration

Parameters
name - the name of a community set
extended - specifies that the referenced community set is of the extended type
delete - removes a set of communities or extended communities from a route
additive - adds a set of communities or extended communities to a route. Because this is the default action, specifying this is optional.
overwrite - specifies that the community set overwrites any communities received with the route

Description
Use the set community-set command to specify a community set in a route map and whether that community set should be added to a route, deleted from a route, or overwritten. The negative form of this command, no set community-set, removes the configured list from the actions to take in this route map.

Notes:
- This command applies only to BGP.
- Specifying an argument for the community set name in the no form of this command has no effect on the configuration. Thus, delete, additive, and overwrite are displayed above as optional.

Default
Community sets are not explicitly added to this route map by default. If one is configured, however, the default action is additive.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route being evaluated against route map "abc" matches interface "fxp1" and the AS path list labeled "list1", then the following set actions will be executed:

- Communities specified in community set "com-set-1" will be added to the matching route.
- Communities specified in community set "com-set-2" will be deleted from the matching route.
- The metric for routes will be set to 50.

    (config)# route-map abc
    (config-route-map)# match interface fxp1
    (config-route-map)# match as-path-list list1
    (config-route-map)# set community-set com-set-1 additive
    (config-route-map)# set community-set com-set-2 delete
    (config-route-map)# set metric 50
    (config-route-map)# exit
    (config)#
set dampen-flap

Name
set dampen-flap - specifies the name of a route flap dampening configuration to be matched

Syntax

```
set dampen-flap name
no set dampen-flap name
```

Mode
Route Map Configuration

Parameters

```
name - a string of characters specifying a dampen flap
```

Description
In order for any damping to take place, routes must match against a route map that specifies damping parameters. The `set dampen-flap` command specifies a dampen flap that contains these parameters.

Default
The `set dampen-flap` command is not explicitly configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if a route matches AS path list "list1", then damping will occur as specified by the parameters in the dampen flap named rfd1.

```
(config)# route-map abc
(config-route-map)# match as-path-list list1
(config-route-map)# set dampen-flap rfd1
(config-route-map)# exit
(config)#
```

set ip next-hop

Name
set ip next-hop - specifies the nexthop with which a BGP route is to be advertised

Syntax

```
set ip next-hop ipv4_address
no set ip next-hop ipv4_address
```

Mode
Route Map Configuration

Parameters

```
ipv4_address - a valid IPv4 address in dotted-quad format
```

Description

Use the `set ip next-hop` command to specify the nexthop with which a BGP route is to be advertised.

Note: This command only applies to BGP. It will be ignored for all other protocols.

Default

BGP will not modify the nexthop value by default.

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if a route matches AS path list "list1", then the nexthop when the route is announced by BGP will be 192.10.2.1.

```
(config)# route-map abc
(config-route-map)# match as-path-list list1
(config-route-map)# set ip next-hop 192.10.2.1
(config-route-map)# exit
(config)#
```

set local-preference

Name

`set local-preference` - sets a route's local preference

Syntax

```
set local-preference num
no set local-preference num?
```

Mode

Route Map Configuration

Parameters

`num` - an integer from 0 to 4,294,967,295, inclusive

Description

Use the `set local-preference` command to set a route's BGP Local Preference value. The negative form of this command, `no set local-preference`, removes the configured `num` value.

Notes:

- This command applies only to BGP routes. It will be ignored for all other routes.
- Specifying a value for `num` in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.

Default

This command is not explicitly configured by default.

Command History

NGC 2.2 - This command was introduced.
Examples
In the following example, if the route matches the AS Path list labeled "aspathlist1", then the local preference value will be set to 10.

```
(config)# route-map abc
(config-route-map)# match as-path-list aspathlist1
(config-route-map)# set local-preference 10
(config-route-map)# exit
```

**set med**

**Name**
`set med` - specifies the MED with which a matching route will be announced

**Syntax**
```
set med [ +value | +igp | value | igp ]
no set med [ +value | +igp | value | igp ]
```

**Mode**
Route Map Configuration

**Parameters**
- +value - an integer to be added to the route’s current MED. This value can be from 0 to 4,294,967,295, inclusive.
- value - an integer to be used as the route’s MED. This value can be from 0 to 4,294,967,295, inclusive.
- +igp - specifies to add the route’s IGP metric to the MED
- igp - specifies to use the route’s IGP metric as the MED

**Description**
Use the `set med` command to change or configure a MED value with which a route will be announced if a match occurs in this route map. **Note:** This command only applies to BGP. It will be ignored for all other protocols.

**Default**
The MED value is not configured by default.

**Command History**
NGC 2.2 - This command was introduced.

**Examples**
In the following example, if the route matches the AS Path list labeled "aspathlist1", then the MED will be set to 1.

```
(config)# route-map abc
(config-route-map)# match as-path-list aspathlist1
(config-route-map)# set med 1
(config-route-map)# exit
(config)#
```
set metric

Name
set metric - specifies the metric with which a matching route will be announced

Syntax
set metric value
no set metric value

Mode
Route Map Configuration

Parameters
value - an integer

Description
Use the set metric command to change or configure a metric with which a route will be announced if a match occurs in this route map. Note: This command has no effect on BGP routes.

Default
The metric value is not configured by default.

Command History
NGC 2.2 - This command was introduced.

Examples
In the following example, if the route matches the AS Path list labeled "aspathlist1", then the metric will be set to 1.

    (config)# route-map abc
    (config-route-map)# match as-path-list aspathlist1
    (config-route-map)# set metric 1
    (config-route-map)# exit
    (config)#

set metric-type

Name
set metric-type - configures the metric types for OSPF and BGP

Syntax
set metric-type [internal | type-1 | type-2]
no set metric-type [internal | type-1 | type-2]

Mode
Route Map Configuration

Parameters
internal - specifies that BGP should use IGP metrics as the MED value
**Route Maps**

**set metric-type**

**Description**

Use the `set metric-type` command to configure metric types for OSPF and BGP. The meanings are specific to each protocol.

**Note:** You cannot include more than one parameter for a single `set metric-type` command

**Default**

If `set metric-type` is not specified, it is the same as if the user had specified the following:

- **for OSPF:** `(config-route-map)# set metric-type type-2`
- **for BGP:** `(config-route-map)# no set metric-type internal`

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if routes are announced with a tag value of 4, then when this route map is imported into a protocol, the metric-type for OSPF routes will be configured as type-2.

```
(config)# route-map abc
(config-route-map)# match tag 4
(config-route-map)# set metric-type type-2
(config-route-map)# exit
(config)#
```

**set origin**

**Name**

`set origin` - sets the origin attribute on matching routes to be advertised to BGP

**Syntax**

```
set origin [egp | igp | incomplete]
no set origin [egp | igp | incomplete]
```

**Mode**

Route Map Configuration

**Parameters**

- **egp** - specifies routes learned from the EGP protocol. This path is most likely not complete.
- **igp** - specifies routes learned from an Intra-Domain Routing Protocol. This path is most likely complete.
- **incomplete** - specifies routes learned from a source other than IGP and EGP

**Description**

Use the `set origin` command to configure the origin attribute on matching routes to be advertised to BGP when this route is exported. An origin of `igp` indicates the route was learned from an Intra-Domain Routing Protocol and is most likely complete. An origin of `egp` indicates the route was learned from the EGP protocol, and the path is most likely not complete. When the route is learned from another source, an origin of `incomplete` is used.
Notes:

- This command applies only to BGP.
- Some competitors set the BGP ORIGIN attribute differently on routes configured via the static statement (origin incomplete) than via the BGP network statement (origin IGP). Advanced Routing Suite always sets origin incomplete. Use the "set origin" ("set origin" on page 483) Route Map command to modify the origin.

**Default**

The set origin command is not configured by default.

**Command History**

NGC 2.2 - This command was introduced.

**Examples**

In the following example, if a route matches AS Path list "as-path-list1", then the numbers 655 will be prepended to the route’s AS_PATH before announcement, and the origin will be set to "incomplete".

```
(config)# route-map xyz
(config-route-map)# match as-path-list as-path-list1
(config-route-map)# set as-path prepend 655
(config-route-map)# set origin incomplete
(config-route-map)# exit
(config)#
```

**set preference**

**Name**

set preference - configures the Advanced Routing Suite preference on routes matching this route map

**Syntax**

```
set preference pref
no set preference pref
```

**Mode**

Route Map Configuration

**Parameters**

pref - an integer between 1 and 255, inclusive

**Description**

Use the set preference command to configure the Advanced Routing Suite preference on routes that match this route map.

**Default**

The default preference value is the preference value of the protocol.

**Command History**

NGC 2.2 - This command was introduced.
Examples

In the following example, if a route’s AS Path matches the AS Path list labeled "aspathlist1", then the preference will be set to 50.

```
(config)# route map abc
(config-route-map)# match as-path-list aspathlist1
(config-route-map)# set preference 50
(config-route-map)# exit
(config)#
```

**set propagate**

**Name**
set propagate - specifies whether type-7 (nssa) routes get translated into type-5 routes on the backbone

**Syntax**

```
set propagate
no set propagate
```

**Mode**
Route Map Configuration

**Parameters**
none

**Description**
Use the set propagate command to specify whether type-7 (nssa) routes get translated into type-5 routes on the backbone. If the flag is not set on a route, then it is not translated.

**Note:** This command applies only to OSPF. It will be ignored for all other protocols.

**Default**
If set propagate is not specified, it is the same as if the user had specified the following:

```
(config-route-map)# no set propagate
```

**Command History**
NGC 2.2 - This command was introduced.

**Examples**

In the following example, if the route being evaluated against route map "abc" matches the BGP Community List "commlist1" and the AS path list labeled "aspathlist1", then exported routes will be propagated.

```
(config)# route-map abc
(config-route-map)# match community commlist1
(config-route-map)# match as-path-list aspathlist1
(config-route-map)# set propagate
(config-route-map)# exit
(config)#
```
set ribs

Name
set ribs specifies the RIBs into which a route will be imported or from which a route will be exported.

Syntax
```
set ribs [ unicast | multicast ]
no set ribs [ unicast | multicast ]
```

Mode
Route Map Configuration

Parameters
- unicast specifies unicast RIB
- multicast specifies multicast RIB

Description
Use the set ribs command to specify the RIBs into which a route will be imported or from which a route will be exported.

Default
By default, a route map applies to all RIBs.

Command History
NGC 2.2 - This command was introduced.

Examples
The following example causes routes in the unicast RIB to be exported into both the unicast and multicast RIBs.
```
(config)# route-map abc
(config-route-map)# match ribs unicast
(config-route-map)# set ribs unicast
(config-route-map)# set ribs multicast
(config-route-map)# exit
(config)#
```

set tag

Name
set tag specifies the tag value with which routes will be announced into OSPFASE, RIP, or RIPng.

Syntax
```
set tag tag_value
no set tag tag_value?
```

Mode
Route Map Configuration
Parameters

*tag_value* – a 32-bit value for the tag specified using a decimal (base 10) number

Description

Use the `set tag` command to propagate tag data from an exterior gateway protocol (such as BGP) through OSPFv2, RIP, or RIPng. The negative form of this command, `no set tag`, removes the configured *tag_value* and returns this to its default value of 0.

Notes:

- Specifying a value for *tag_value* in the `no` form has no effect on the configuration. Thus, it is displayed above as optional.
- This command will be ignored for any protocol that is not OSPFv2, RIP, or RIPng.

Default

If `set tag` is not specified, it is the same as if the user had specified the following:

```
(config-route-map)# set tag 0
```

Command History

NGC 2.2 - This command was introduced.

Examples

In the following example, if routes are received with a tag value of 4, then when a matching route is imported into a protocol, the preference for that route will be configured as 50, and the new tag value will be 5.

```
(config)# route-map abc
(config-route-map)# match tag 4
(config-route-map)# set preference 50
(config-route-map)# set tag 5
(config-route-map)# exit
(config)#
```
b dip sends-group-always • 93
b dip tie-break-on-age • 94
Border Gateway Protocol (BGP) • 71
Border Gateway Protocol (BGP) Overview • 73

c clear ip bgp • 94
clear ip igmp group • 333
clear ip mrout • 63
CLI Behavior Commands • 23
CLI Modes • 21
Command History • 16
Command Line Completion • 19
Command Tokens • 19
compatible rfc1583 • 176
configure file • 23
configure terminal • 23
Context-Sensitive Help • 20
cost • 227

d dampen-flap • 449
dead-interval • 176, 228
Default • 16, 18
default-metric • 95, 219, 281
default-nssa-metric • 219
default-nssa-type • 220
default-preference • 221
default-tag • 222
default-type • 222
Description • 16, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 37
disable • 24, 43
Disabling/Enabling CLI Tracing • 21
distance • 96, 177
Distance Vector Multicast Routing Protocol
(DVMRP) • 317
Distance Vector Multicast Routing Protocol
(DVMRP) Overview • 317
distribute-list • 97, 282
Documentation of Commands • 14

e ecmp • 285
enable • 24, 98, 178, 229, 285
enable-te • 179
end • 25
Example 1 • 28
Example 2 • 28
Example 3 • 297, 311
Examples • 16, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 37
exit • 25
Exiting the CL • 21

f Fast Open Shortest Path First (OSPF) • 166
Fast Open Shortest Path First (OSPF)
Overview • 168
Field Descriptions • 33, 37
flash-update-time • 286

g General Concepts • 39