

TUVWXYZ

t1 bert

- Description:** Enables bit error rate tests using the specified pattern on a T1 line on a CT3 module. The **no** version stops the test that is running.
- Syntax:** `t1 channel bert pattern pattern interval time [unframed]`
`no t1 channel bert`
- *channel* – T1 channel number in the range 1–28
 - *pattern* – one of the following test patterns
 - › 0s – repetitive test pattern of all zeros, 00000...
 - › 1s – repetitive test pattern of all ones, 11111...
 - › 2^11 – pseudorandom test pattern, 2048 bits in length
 - › 2^15 – pseudorandom 0.151 test pattern, 32768 bits in length
 - › 2^20-O153 – pseudorandom 0.153 test pattern, 1048575 bits in length
 - › 2^20-QRSS – pseudorandom QRSS 0.151 test pattern, 1048575 bits in length
 - › 2^23- – pseudorandom 0.151 test pattern, 8388607 bits in length
 - › alt-0-1 – repetitive alternating test pattern of zeros and ones, 01010101...
 - › *time* – interval between test patterns, ranging from 1–14,400 minutes
 - unframed – the test bit pattern occupies all bits on the link, overwriting the framing bits. If you do not specify the unframed keyword, then the test bit pattern only occupies T1 payload bits.
- Mode(s):** Controller Configuration

t1 clock source

- Description:** Determines which end of the T1 interface provides clocking. The **no** version uses the default value, **line**.
- Syntax:** t1 *channel* clock source { line | internal { module | chassis } }
no t1 *channel* clock source
- *channel* – T1 channel number in the range 1–28
 - line – interface transmits data from a clock recovered from the line's receive data stream
 - internal – interface transmits data using its internal clock. You must specify one of the following for internal clocking:
 - › module – internal clock is from the line module itself
 - › chassis – internal clock is from the configured system clock
- Mode(s):** Controller Configuration

t1 fdl

- Description:** Specifies the FDL standard used by a specific T1 channel on the CT3 interface. The **no** version disables FDL support.
- Syntax:** t1 *channel* fdl { ansi | att | all | none }
no t1 *channel* fdl [ansi | att | all]
- *channel* – T1 channel number in the range 1–28
 - ansi – specifies ANSI T1.403 for extended superframe FDL exchange support
 - att – specifies AT&T technical reference 54016 for extended superframe FDL exchange support
 - all – specifies both the AT&T and ANSI mode for extended superframe FDL exchange support
 - none – removes the current FDL mode settings
- Mode(s):** Controller Configuration

t1 fdl carrier

- Description:** Specifies that an interface is used in the carrier environment of a specific T1 channel on the CT3 interface. The **no** version restores the default situation, in which an interface does not operate in the carrier environment. Currently, the only valid value is **no t1 fdl carrier**.
- Syntax:** [no] t1 *channel* fdl carrier
- *channel* – T1 channel number in the range 1–28
- Mode(s):** Controller Configuration

t1 fdl string

- Description:** Allows you to configure an FDL message as defined in the ANSI T1.403 specification on a specific T1 channel on the CT3 interface. Currently, FDL strings can only be configured locally. The **no** version disables the FDL messages.
- Syntax:** t1 *channel* fdl string { eic *eicValue* | fic *ficValue* | lic *licValue* | unit *unitValue* }
no t1 *channel* fdl string { eic | fic | lic | unit }
- *channel* – T1 channel number in the range 1–28
 - *eicValue* – equipment identification code; 1–10 characters; factory default is disabled
 - *licValue* – location identification code; 1–10 characters; factory default is disabled
 - *ficValue* – frame identification code; 1–10 characters; factory default is disabled
 - *unitValue* – unit identification code; 1–6 characters; factory defaults:
 - › For T3 lines is slot/controller/t1 in SSCCLL format. For example: t1 5 on controller t3 12/2 is represented: 120205
 - › For T1 lines is slot/controller in AASSCC format. For example: controller t1 4/2 is represented: AA0402
- Mode(s):** Controller Configuration

t1 framing

- Description:** Specifies the type of framing used by a specific T1 channel on the CT3 interface. The **no** version uses the default value, **esf**.
- Syntax:** t1 *channel* framing { esf | sf }
no t1 *channel* framing
- *channel* – T1 channel number in the range 1–28
 - esf – specifies extended superframe
 - sf – specifies superframe
- Mode(s):** Controller Configuration

t1 lineCoding

- Description:** Specifies the type of line coding used by a specific T1 channel on the CT3. The **no** version uses the default value, **b8zs**.
- Syntax:** t1 *channel* lineCoding { ami | b8zs }
no t1 *channel* lineCoding
- *channel* – T1 channel number in the range 1–28
 - ami – specifies alternate mark inversion
 - b8zs – specifies bipolar with eight-zero substitution
- Mode(s):** Controller Configuration

t1 loopback

Description: Configures a loopback test for a T1 line on a CT3 module. The **no** version deactivates the loopback test; if you specify the **remote** keyword, the **no** version sends the 16-bit ESF data link code word or inband pattern to deactivate the loopback at the remote end based on the last activate request sent to the remote end. If you do not specify the **remote** keyword, the **no** version clears the local loopback configuration.

Syntax: t1 *channel* loopback [local | network { line | payload } | remote { line { fdl { ansi | bellcore } | inband } payload [fdl] [ansi] }]

no t1 *channel* loopback [remote]

- *channel* – T1 channel number in the range 1–28
- *local* – loops the router output data back toward the router at the T1 framer and sends an alarm indication signal out toward the network. This is the default setting if you specify no optional keywords.
- *network { line | payload }* – Specify the *line* keyword to loop the data back toward the network before the T1 framer and automatically set a local loopback at the HDLC controllers. Specify the *payload* keyword to loop the payload data back toward the network at the T1 framer and automatically set a local loopback at the HDLC controllers.
- *remote line fdl ansi* – sends a repeating 16-bit ESF data link code word (00001110 11111111) to the remote end requesting that it enter into a network line loopback. Specify the *ansi* keyword to enable the remote line FDL ANSI bit loopback on the T1 channel, according to the ANSI T1.403 specification.
- *remote line fdl bellcore* – sends a repeating 16-bit ESF data link code word (00010010 11111111) to the remote end requesting that it enter into a network line loopback. Specify the *bellcore* keyword to enable the remote line FDL Bellcore bit loopback on the T1 channel, according to the Bellcore TR-TSY-000312 specification.
- *remote line inband* – sends a repeating 5-bit inband pattern (00001) to the remote end requesting that it enter into a network line loopback
- *remote payload [fdl] [ansi]* – sends a repeating 16-bit ESF data link code word (00010100 11111111) to the remote end requesting that it enter into a network payload loopback. Enables the remote payload FDL ANSI bit loopback on the T1 channel. You can optionally specify *fdl* and *ansi*, but it is not necessary.

Mode(s): Controller Configuration

t1 remote-loopback

- Description:** Enables the acceptance of remote loopback requests. The **no** version restores the default value, which is to reject remote loopback requests.
- Syntax:** [no] t1 *channel* remote-loopback
- *channel* – T1 channel number in the range 1–28
- Mode(s):** Controller Configuration

t1 shutdown

- Description:** Disables a T1 or fractional T1 channel on the CT3. The **no** version restarts a disabled interface.
- Syntax:** [no] t1 *channel* | *channel/subchan* shutdown
- *channel* – T1 channel number in the range 1–28
 - *subchan* – FT1 subchannel on a T1 interface in the range 1–24
- Mode(s):** Controller Configuration

t1 snmp trap link-status

- Description:** Enables processing of SNMP link status information on a T1 or fractional T1 channel on the CT3. The **no** version disables the processing of SNMP link status information.
- Syntax:** [no] t1 *channel* | *channel/subchan* snmp trap link-status
- *channel* – T1 channel number in the range 1–28
 - *subchan* – specifies the FT1 subchannel on a T1 interface in the range 1–24
- Mode(s):** Controller Configuration

t1 timeslots

- Description:** Configures the timeslots and data rate used on each T1 channel on the CT3 interface. The **no** version deletes the fractional T1 circuit.
- Syntax:** `t1 channel/subchan timeslots range [speed { 56 | 64 }]`
`no t1 subchan`
- *channel* – T1 channel number in the range 1–28
 - *subchan* – subchannel specifies the logical subchannel on a T1 in the range 1–24
 - *range* – specifies the timeslot assigned to the T1 channel in the range 1–24. A dash represents a range of timeslots, and a comma separates timeslots. For example, 1-10, 15-18 assigns timeslots 1 through 10 and 15 through 18.
 - *speed* – specifies the data rate for the T1 channel. Values are 56 Kbps or 64 Kbps. The default is 64 Kbps.
- Mode(s):** Controller Configuration

t1 yellow

- Description:** Generates or detects a yellow alarm for a T1 channel on the CT3 interface. The **no** version restores the default value, to not generate or to not detect a yellow alarm.
- Syntax:** `[no] t1 channel yellow { generate | detect }`
- *channel* – T1 channel number in the range 1–28
- Mode(s):** Controller Configuration

table-map

- Description:** Applies the specified route map to all BGP routes about to be added to the IP routing table. The **no** version halts application of the route map.
- Syntax:** `table-map mapTag`
`no table-map [mapTag]`
- *mapTag* – name of the route map; a string of up to 32 alphanumeric characters
- Mode(s):** Address Family Configuration, Router Configuration

tag

- Description:** Specifies a user-defined tag. The **no** version removes the tag from the operation. You can configure a tag for both echo and echoPath types.
- Syntax:** tag *tagValue*
no tag
- *tagValue* – name of a group that the operation belongs to: 0–255 ASCII characters; the default is to have no tag
- Mode(s):** RTR Configuration

telnet

Description: Enables connections to remote systems via the embedded Telnet client. There is no **no** version.

Syntax: `telnet IpAddress | hostname [vrf vrfName] [ipPortNumber | ipPortType] [source-interface interfaceType interfaceSpecifier | noecho | line | debug | verbose]*`

- *IpAddress* – IP address of the remote system
- *hostname* – name of the remote system
- *vrfName* – name of the VRF to which the command applies; string of 1–32 alphanumeric characters
- *ipPortNumber* – number in the range 0-65535 of the port for the connection to the remote system. The default is port number 23, the Telnet port. For more information on port numbers and associated processes, see www.iana.org.
- *ipPortType* – name of a well-known port, as follows:
 - › *bgp* – Border Gateway Protocol (port 179)
 - › *chargen* – character generator (port 19)
 - › *cmd* – remote commands (port 514)
 - › *daytime* – daytime (port 13)
 - › *discard* – discard (port 9)
 - › *domain* – Domain Name Service (port 53)
 - › *echo* – echo (port 7)
 - › *exec* – exec (port 512)
 - › *finger* – finger (port 79)
 - › *ftp* – File Transfer Protocol (port 21)
 - › *ftp-data* – FTP data connections (port 20)
 - › *gopher* – gopher (port 70)
 - › *hostname* – NIC hostname server (port 101)
 - › *ident* – Ident Protocol (port 113)
 - › *irc* – Internet Relay Chat (port 194)
 - › *klogin* – Kerberos login (port 543)
 - › *kshell* – Kerberos shell (port 544)
 - › *login* – Login (port 513)
 - › *lpd* – printer service (port 515)
 - › *nntp* – Network News Transport Protocol (port 119)

- › pim-auto-rp – Protocol Independent Multicast Auto RP (port 496)
- › pop2 – Post Office Protocol version 2 (port 109)
- › pop3 – Post Office Protocol version 2 (port 110)
- › smtp – Simple Mail Transport Protocol (port 25)
- › sunrpc – Sun Remote Procedure Call (port 111)
- › syslog – Syslog (port 514)
- › tacacs – Terminal Access Concentrator Access Control System (port 49)
- › talk – Talk (port 517)
- › telnet – Telnet (port 23)
- › time – Time (port 37)
- › uucp – Unix-to-Unix Copy Program (port 540)
- › whois – nickname (port 43)
- › www – World Wide Web (port 80)
- source-interface – forces Telnet to use the IP address of the specified interface as the source address for the Telnet connection
 - › *interfaceType* – type of interface to use to obtain the source address for the Telnet connection; see *Interface Types and Specifiers* in *About This Guide*
 - › *interfaceSpecifier* – number of interface to use to obtain the source address for the Telnet connection; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
- noecho – disables local echo of user input
- line – enables line mode
- debug – enables debugging
- verbose – enables verbose mode
- * – indicates that one or more parameters can be repeated multiple times in a list in the command line

Mode(s): Privileged Exec

telnet listen

- Description:** Sets the Telnet daemon to listen in a virtual router other than the default. The **no** version deletes the Telnet daemon.
- Syntax:** telnet listen [port *portValue*]
no telnet listen
- *portValue* – TCP port on which the Telnet daemon listens; if not specified, the default port 23 is used
- Mode(s):** Global Configuration

terminal data-character-bits

- Description:** Sets the number of data bits available for characters for the current session on the terminal screen. There is no **no** version.
- Syntax:** terminal data-character-bits { 7 | 8 }
- 7 – 7 data bits per character; this setting supports only characters in the standard ASCII set
 - 8 – 8 data bits per character; default setting, supports the full set of 8-bit international characters
- Mode(s):** User Exec, Privileged Exec

terminal length

- Description:** Sets the number of lines on the current terminal screen for the current session. There is no **no** version.
- Syntax:** terminal length *value*
- *value* – number for the screen length in the range 0–512. If 0, the router does not pause between screens of output. If not 0, the router pauses between screens.
- Mode(s):** User Exec, Privileged Exec

terminal speed

- Description:** Sets the speed for the current console session. There is no **no** version.
- Syntax:** terminal speed *baudRate*
- *baudRate* – terminal speed for the current console session; possible values are: 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200
- Mode(s):** Privileged Exec

terminal width

- Description:** Sets the number of character columns on the current terminal screen for the current line for a session. There is no **no** version.
- Syntax:** terminal width *value*
- *value* – number of characters in the range 30–512
- Mode(s):** User Exec, Privileged Exec

test aaa

- Description:** Verifies RADIUS authentication and accounting and IP address assignment setup. The test uses a username and password and attempts to authenticate a user, get an address assignment, and issue a start accounting request. The test immediately terminates the session by issuing a stop accounting request and an address release. Optionally, a virtual router context may be specified.
- Syntax:** test aaa { ppp | mlppp } *userName* [*password*] [virtual-router *vrContext*]
- ppp – indicates a PPP user
 - mlppp – simulates Multilink PPP
 - *userName* – username to test
 - *password* – password to associate with username; password is optional—when omitted, the RADIUS access request has no User-Password attribute
 - *vrContext* – virtual router context in which to authenticate the user
- Mode(s):** Privileged Exec

threshold

- Description:** Sets the threshold values for bit error rates used in APS/MSP alarms. The **no** version restores the default value, 6, for the specified alarm.
- Syntax:** threshold { sd-ber | sf-ber } *rate*
- no threshold { sd-ber | sf-ber }
- sd-ber – bit error rate that specifies signal degradation
- sf-ber – bit error rate that specifies signal failure
- rate* – integer in the range 3–9; a value of *n* corresponds to a rate of 10^{-n} (10e-*n*) errors per second
- Mode(s):** Controller Configuration

timeout

Description: When used from RADIUS Configuration mode, specifies the interval, in seconds, before the system retransmits a RADIUS packet to an authentication or accounting server. The **no** version uses the default.

When used from RTR Configuration mode, specifies the timeout for a Response Time Reporter operation. The **no** version returns the operation to the default value. You can apply this parameter only to *echo* entries.

Syntax: RADIUS:

timeout waitTime

no timeout

- *waitTime* – specifies the number of seconds in the range 3–10. The default is 3.

RTR:

timeout timeoutValue

no timeout

- *timeoutValue* – number in milliseconds that the operation waits for a response; if the value is set to 0 or is larger than frequency, it will be ignored; the default is 5000 milliseconds

Mode(s): Radius Configuration, RTR Configuration

timeout login response

Description: Sets a time limit during which users must provide a password when they log into the console or a vty line. Specifying a value of 0 indicates that there is no time limit during which users must enter a password. The **no** version restores the default value, 30 seconds.

Syntax: *timeout login response seconds*

no timeout login response

- *seconds* – length of the timeout in the range 0–300 seconds

Mode(s): Line Configuration

timers

- Description:** Configures RIP timers. The **no** version restores the default values.
- Syntax:** `timers update invalid holddown flush`
- `no timers`
- *update* – interval in seconds at which routing updates are sent. The default is 30.
 - *invalid* – interval in seconds after which a route is declared invalid (null). The default is 180.
 - *holddown* – interval in seconds during which routing information regarding better paths is disregarded by the system. The default is 120.
 - *flush* – interval in seconds that must pass before a route is removed from the routing table. Set this value greater than the invalid value. The default is 300.
- Mode(s):** Router Configuration

timers bgp

- Description:** Sets keepalive and hold-time timers for all neighbors. The **no** version restores the default values.
- Syntax:** `timers bgp keepaliveTime holdTime`
- `no timers bgp [keepaliveTime [holdTime]]`
- *keepaliveTime* – interval in seconds between keepalive messages; range is 0–65535 seconds; default is 30 seconds; a value of zero prevents BGP from sending keepalive messages
 - *holdTime* – period in seconds that BGP waits for keepalive messages before declaring the neighbor to be unavailable; range is 0–65535 seconds; default is 90 seconds; a value of zero informs BGP not to expect any hold-time messages
- Mode(s):** Router Configuration

timers spf

- Description:** Configures the delay time between when OSPF receives a topology change and when it starts an SPF calculation and the hold time between two consecutive SPF calculations. The **no** version restores the default value.
- Syntax:** `[no] timers spf holdTime`
- *holdTime* – number in the range 1–5 seconds; default is 3seconds; the hold time between consecutive SPF calculations
- Mode(s):** Router Configuration

time-to-live

- Description:** Specifies a hop count by setting the value of the time-to-live field used by packets sent to a RIP remote neighbor. The **no** version restores the default value.
- Syntax:** time-to-live *ttlValue*
no time-to-live
- *ttlValue* – number in the range 1–16; the default value is 16
- Mode(s):** Remote Neighbor Configuration

timing disable-auto-upgrade

- Description:** Disables the auto-upgrade feature of the system timing. The **no** version enables the auto-upgrade feature.
- Syntax:** [no] timing disable-auto-upgrade
- Mode(s):** Global Configuration

timing select

- Description:** Configures the preferred timing selector. There is no **no** version.
- Syntax:** timing select *selector*
- *selector* – timing selector
 - › primary – highest-priority preferred selection
 - › secondary – middle-priority preferred selection
 - › tertiary – lowest-priority preferred selection
- Mode(s):** Global Configuration

timing source

- Description:** Configures the system's timing sources. Only one of these timing sources can be an external source received via an interface on an I/O module other than the SRP I/O module; the other two must be either internal sources or external sources received via the SRP I/O modules. There is no **no** version.
- Syntax:** `timing source selector { internal | line lineType | { sonet | ds3 | ds1 | e1 | e3 } interfaceValue }`
- *selector* – priority of the timing source; in descending order **primary**, **secondary**, or **tertiary**
 - *internal* – internal SC oscillator
 - *line* – external timing input on SRP module
 - *lineType* – one of the following timing sources:
 - › e1:a – E1 clock, port A on SRP module
 - › e1:b – E1 clock, port B on SRP module
 - › t1:a – T1 clock, port A on SRP module
 - › t1:b – T1 clock, port B on SRP module
 - *sonet* – specifies a SONET interface
 - *ds3* – specifies a DS3 interface
 - *ds1* – specifies a DS1 interface
 - *e1* – specifies an E1 interface
 - *e3* – specifies an E3 interface
 - *interfaceValue* – interface specifier, in the form slot/port[:subPort]
- Mode(s):** Global Configuration

tos

- Description:** Defines a type of service byte in the RTR operation's IP header. The **no** version returns the operation to the default value.
- Syntax:** `tos tosValue`
- `no tos`
- *tosValue* – ToS byte in the IP header: 0–255; the default is 0 for both RTR types.
- Mode(s):** RTR Configuration

traceroute

Description: Discovers the paths that router packets follow when travelling to their destinations. There is no **no** version.

Syntax: `traceroute [vrf vrfName] destination [ttl maxTTLCount]
 [timeout timeOutVal] [data-size sizeValue]
 [source { interface interfaceType interfaceSpecifier |
 address sourceAddress }]
 [extended [tos tosVal] [set-dont-fragment-bit] [interface iType iNumber]`

- *vrfName* – name of the VRF context
- *destination* – IP address or domain name of the trace
- *maxTTLCount* – maximum number of hops of the trace in the range 1–255; default is 32
- *timeOutVal* – time in seconds to wait for trace responses in the range 1–20; default is 2
- *sizeValue* – number of bytes comprising the IP packet and reflected in the IP header in the range 0–64000
- source interface – specifies an interface as the source for the transmitted packets
 - › *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
 - › *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
- source address – specifies an IP address as the source for the transmitted packets
 - › *sourceAddress* – IP address or domain name used as the source address
- extended – specifies extended IP header attributes
- *tosVal* – value of the ToS byte
- set-dont-fragment-bit – specifies the don't-fragment bit
- *iType* – interface type
- *iNumber* – interface location

Mode(s): User Exec, Privileged Exec

traffic-class

- Description:** In Policy Configuration mode, specifies a traffic class in a policy list for policy management. The **no** version removes a traffic class from a policy list. The **suspend** version temporarily suspends the policy rule.
- In Global Configuration mode, configures a traffic class in the ERX system. In Traffic Class Group Configuration mode, specifies a traffic class that belongs to the traffic class group. The **no** version deletes the traffic class.
- Syntax:** In Policy Configuration mode:
- ```
[no] [suspend] traffic-class trafficClassName [classifier-group clacName]
[precedence precValue]
```
- *trafficClassName* – name for the traffic class
  - *clacName* – name for the classifier group
  - *precValue* – precedence value in the range 0-32768
- In Global Configuration and Traffic Class Group Configuration modes:
- ```
[ no ] traffic-class trafficClassName
```
- *trafficClassName* – name for the traffic class
- Mode(s):** Global Configuration, Policy Configuration, Traffic Class Group Configuration

traffic-class-group

- Description:** Configures a traffic class group. The **no** version deletes the selected traffic-class group.
- Syntax:** [no] traffic-class-group *trafficClassGroupName*
- *trafficClassGroupName* – name for the traffic class group
- Mode(s):** Global Configuration

transmit-delay

- Description:** Sets the estimated time it takes to transmit a link state update packet on the OSPF remote-neighbor interface. The **no** version restores the default value.
- Syntax:** transmit-delay *transmDelay*
- no transmit-delay
- *transmDelay* – the link state transmit delay, a number in the range 0–3600 seconds; default value is 1 second
- Mode(s):** Remote Neighbor Configuration

ttl

Description:	Specifies a hop count by setting the value of the time-to-live field used by packets sent to an OSPF remote neighbor. The no version restores the default value.
Syntax:	ttl <i>ttlValue</i> no ttl <ul style="list-style-type: none"> • <i>ttlValue</i> – number in the range 1–255; the default value is 1
Mode(s):	Remote Neighbor Configuration

tunnel

Description:	Specifies an L2TP or L2F tunnel and changes the mode to Domain Map Tunnel Configuration. In Domain Map Tunnel Configuration mode, you can set the attributes of the tunnel. The no version deletes the L2TP or L2F tunnel configuration from the system.
Syntax:	[no] tunnel <i>tag</i> <ul style="list-style-type: none"> • <i>tag</i> – tunnel tag value in the range 1–31
Mode(s):	Domain Map Configuration

tunnel checksum

Description:	Enables end-to-end checksum computation for GRE tunnels. The no version disables the checksum option.
Syntax:	[no] tunnel checksum
Mode(s):	Interface Configuration

tunnel destination

Description:	For DVMRP or GRE, configures the tunnel endpoint for static tunnels. The no version deletes the endpoint. For IPsec, configures the remote tunnel endpoint. For MPLS in Interface Configuration mode, configures the tunnel endpoint for static MPLS tunnels. The no version deletes the endpoint. For MPLS in Tunnel Profile Configuration mode, configures the source of tunnel endpoints (destinations) within a tunnel profile. You can specify that the endpoints are to be learned from BGP, IS-IS, or OSPF, or you can provide one or more IP addresses as the endpoint(s). If you specify the destination address, it must be the address of the MPLS interface or the router ID of the destination router. The no version deletes the endpoints.
---------------------	---

- Syntax:** For DVMRP and GRE:
- tunnel destination *ipAddress* | *hostname*
- no tunnel destination
- *ipAddress* – IP address of the interface on the remote system
 - *hostname* – name of the host that will serve as the tunnel endpoint
- For IPSec:
- tunnel destination *ipAddress*
- no tunnel destination
- *ipAddress* – IP address of the interface on the remote system or the router ID of the destination router that serves as the tunnel endpoint
- For MPLS in Interface Configuration mode:
- tunnel destination *ipAddress* [mask *ipMask*]
- no tunnel destination
- For MPLS in Tunnel Profile Configuration mode:
- [no] tunnel destination
{ *ipAddress* [mask *ipMask*] | *ipAddress* [*ipAddress*]* |
{ isis-level-2 | ospf-bdr } [{ access-list | prefix-list } *listName*] }
- isis-level-2 – specifies IS-IS level-2 routers as acceptable destinations
 - ospf-bdr – specifies OSPF border routers as acceptable destinations
 - *listName* – name of access list or prefix list that contains the IP addresses that are acceptable as tunnel endpoints
 - *ipAddress* – IP address of the interface on the remote system or the router ID of the destination router that serves as the tunnel endpoint; for a tunnel profile, you can list multiple addresses
 - *ipMask* – mask for the destination IP address
 - * – indicates that one or more parameters can be repeated multiple times in a list in the command line
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel ip profile

- Description:** Assigns an IP profile to the MPLS tunnel. The **no mpls tunnels profile** version removes the IP profile from the tunnel.
- Syntax:** tunnel ip profile *ipProfileId*
no tunnel ip profile
- *ipProfileId* – name of an IP profile
- Mode(s):** Tunnel Profile Configuration

tunnel lifetime

- Description:** Sets the lifetime of IPSec SAs running on this tunnel. You can specify the lifetime in seconds and/or volume of traffic. Before either limit is reached, the SA is renegotiated, ensuring that the tunnel does not go down before the renegotiation is finished. The **no** version sets the lifetime to the default lifetime defined by the **ipsec lifetime** command.
- Syntax:** tunnel lifetime { seconds *seconds* kilobytes *kilobytes* | seconds *seconds* | kilobytes *kilobytes* }
no tunnel lifetime { seconds | kilobytes }
- *seconds* – number of seconds security SAs on this tunnel live before expiring; the range is 7200–4284967295
 - *kilobytes* – volume of traffic in kilobytes that can pass between the tunnel endpoints using a given SA before the SA expires; the range is 102400–4294967295
- Mode(s):** Interface Configuration

tunnel local-identity

- Description:** Specifies the local identity of the IPSec tunnel. The **no** version removes the local endpoint and sets the default identity, which is subnet 0.0.0.0 0.0.0.0.
- Syntax:** tunnel local-identity { address *ipAddress* | subnet *ipAddress subnetMask* | range *ipAddressLow ipAddressHigh* }
no tunnel local-identity
- address – specifies an IP address as the local identity
 - subnet – specifies a subnet as the local identity
 - range – specifies a range of IP addresses as the local identity
- Mode(s):** Interface Configuration

tunnel mpls affinity

- Description:** Assigns an affinity to the tunnel. The **no** version removes the affinity from the tunnel.
- Syntax:** tunnel mpls [traffic-eng] affinity *affinity* [mask *mask*]
no tunnel mpls [traffic-eng] affinity
- traffic-eng – optional keyword for compatibility with non-ERX implementations
 - *affinity* – attributes that must be configured on the interface in order to be considered by the tunnel; ranges from 0x0 to 0xFFFFFFFF; the default is 0x0
 - *mask* – mask to identify attributes to be checked; a 1 signifies that the attribute value must match, a 0 signifies that the attribute value does not matter; ranges from 0x0 to 0xFFFFFFFF; the default is 0x0000FFFF
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls autoroute announce

- Description:** Configures the LSP tunnel to register its endpoint (the egress router) with the configured routing protocol. If you do not specify a routing protocol, the default is IS-IS and OSPF. The **no** version disables endpoint announcements.
- Syntax:** [no] tunnel mpls [traffic-eng] autoroute announce [ospf | isis | bgp]
- traffic-eng – optional keyword for compatibility with non-ERX implementations
 - ospf – endpoint is announced to OSPF
 - isis – endpoint is announced to IS-IS
 - bgp – endpoint is announced to BGP
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls autoroute metric

- Description:** Specifies the tunnel metric. The value determines tunnel preference when there is more than one tunnel or native IP path to a tunnel endpoint. A lower value is preferred to a higher value. When you set up multiple tunnels, if the primary tunnel goes down, the existing tunnel with the lowest metric is used immediately. If you specify an absolute value from 1–65535, this value overrides the metric for the path provided by the IGP. If you specify a relative value from -10 to +10, this value is subtracted from (-) or added to (+) the metric for the path provided by the IGP. The **no** version restores the default value of relative 0, meaning that the tunnel metric is the IGP value.
- Syntax:** tunnel mpls [traffic-eng] autoroute metric {absolute | relative} *metricValue*
no tunnel mpls [traffic-eng] autoroute metric
- traffic-eng – optional keyword for compatibility with non-ERX implementations
 - absolute – metric is an absolute value
 - relative – metric is a signed relative value
 - *metricValue* – preference value for a path
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls bandwidth

- Description:** Specifies the bandwidth required for the tunnel. The **no** version removes the bandwidth constraint from the tunnel.
- Syntax:** tunnel mpls [traffic-eng] bandwidth *bandwidth*
no tunnel mpls [traffic-eng] bandwidth
- traffic-eng – optional keyword for compatibility with non-ERX implementations
 - *bandwidth* – amount of bandwidth required for the tunnel in kilobits per second, a value ranging from 0–4294967295; the default value is 0
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls base-tunnel

- Description:** Stacks the tunnel on the specified MPLS base tunnel. The **no** version removes the tunnel from on top of the base tunnel.
- Syntax:** tunnel mpls base-tunnel *baseName*
no tunnel mpls base-tunnel
- *baseName* – name of the tunnel; up to 20 alphanumeric characters
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls description

- Description:** Associates a description with the MPLS tunnel. The **no** version deletes the description.
- Syntax:** tunnel mpls description *textString*
no tunnel mpls description
- *textString* – description or name of the tunnel; up to 40 alphanumeric characters
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls dynamic target

- Description:** Specifies that the tunnel is a targeted tunnel for MPLS to stack above a base LSP that is dynamically created by topology-driven LDP. The **no** version stops the tunnel from being dynamically stacked over a base LSP.
- Syntax:** [no] tunnel mpls dynamic target
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls ip propagate-ttl

- Description:** Controls the value for the TTL field in the MPLS header when a label is assigned to an IP packet. Enabled by default, this command sets the TTL to the TTL value from the IP packet header. Optionally controls how network structure is hidden from the **traceroute** command; by default the structure is revealed to forwarded and local packets. The **no** version sets the value to 255. The **default** version reverts to the settings configured at the global level; if none, reverts to the global default (causing the TTL field to be copied from the IP packet header, enabling the **traceroute** command to show all the hops in the network, and propagating the label TTL into the IP header at the tunnel egress).
- Syntax:** tunnel mpls ip propagate-ttl [forwarded | local]
{ no | default } tunnel mpls ip propagate-ttl [forwarded | local]
- **forwarded** – controls whether the network structure is revealed to or hidden from **traceroute** for forwarded packets
 - **local** – controls whether the network structure is revealed to or hidden from **traceroute** for local packets
- Mode(s):** Interface Configuration

tunnel mpls label-dist

- Description:** Specifies the label distribution protocol. The **no** version removes the label distribution protocol.
- Syntax:** tunnel mpls [traffic-eng] label-dist { cr-ldp | rsvp-te }
no tunnel mpls [traffic-eng] label-dist
- traffic-eng – optional keyword for compatibility with non-ERX implementations
 - cr-ldp – sets the label distribution protocol to CR-LDP
 - rsvp-te – sets the label distribution protocol to RSVP-TE
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls no-route retries

- Description:** Specifies for a particular tunnel the number of attempts that will be made to set up an LSP for CR-LDP and RSVP-TE after a failure due to no available route. The **no** version restores the default value, 0, which means the attempts will be made until successful.
- Syntax:** tunnel mpls lsp no-route retries *retryNum*
no tunnel mpls lsp no-route retries
- *retryNum* – number of retry attempts from 0–65535
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls no-route retry-time

- Description:** Specifies for a particular tunnel the interval in seconds between attempts to set up an LSP for CR-LDP and RSVP-TE after a failure due to no available route. The **no** version restores the default value of 5 seconds.
- Syntax:** tunnel mpls lsp no-route retry-time *retryTime*
no tunnel mpls lsp no-route retry-time
- *retryTime* – interval from 1–60
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mpls path-option

Description: Specifies the path options for a tunnel. You can configure one or more path options—each identified by a unique number—for a given tunnel. The path option number expresses the preference for that option; lower numbers have a higher preference, with 1 having the highest preference. The **no** version deletes the path options.

Syntax: tunnel mpls [traffic-eng] path-option *number*
{ dynamic | explicit { name *pathName* | identifier *idNumber* } }
[hop-by-hop | ospf | isis] [lockdown]

no tunnel mpls [traffic-eng] path-option *number*

- traffic-eng – optional keyword for compatibility with non-ERX implementations
- *number* – identifier for a set of path options
- dynamic – the path is dynamically calculated
- explicit – an explicit path is used
 - › *pathName* – name of the explicit path, a string of up to 20 characters
 - › *idNumber* – number identifying the explicit path, ranging from 1–65535
- hop-by-hop – specifies that hop-by-hop routing is used for this path option
- ospf – specifies that OSPF routing is used for this path option
- isis – specifies that IS-IS routing is used for this path option
- lockdown – specifies that optimization is not done for this path option

Mode(s): Interface Configuration, Tunnel Profile Configuration

tunnel mpls priority

Description: Assigns a setup priority and optionally a hold priority to the tunnel. The priority can range from 0 (the highest) to 7 (the lowest). The hold priority, if set, must be equal to or better (lower numerically) than the setup priority. In the event of insufficient resources when a tunnel is being established, its setup priority is evaluated against the hold priorities of existing tunnels. Tunnels with lower hold priorities (higher values) are preempted and torn down to free their resources for the new tunnel. The **no** version removes the priority from the tunnel.

Syntax: tunnel mpls [traffic-eng] priority *setupPriority* [*holdPriority*]
no tunnel mpls [traffic-eng] priority

- *traffic-eng* – optional keyword for compatibility with non-ERX implementations
- *setupPriority* – priority for the tunnel as it is being established; the default value is 4
- *holdPriority* – priority for the tunnel after it has been established; the default value is equal to the configured value of the setup priority

Mode(s): Interface Configuration, Tunnel Profile Configuration

tunnel mpls retries

Description: Specifies for a particular tunnel the number of attempts that will be made to set up an LSP for CR-LDP and RSVP-TE after a failure other than one due to no available route. The **no** version restores the default value, 0, which means the attempts will be made until successful.

Syntax: tunnel mpls retries *retryNum*
no tunnel mpls retries

- *retryNum* – number of retry attempts from 0–65535

Mode(s): Interface Configuration, Tunnel Profile Configuration

tunnel mpls retry-time

Description: Specifies for a particular tunnel the interval in seconds between attempts to set up an LSP for CR-LDP and RSVP-TE after a failure other than one due to no available route. The **no** version restores the default value of 5 seconds.

Syntax: tunnel mpls [no-route] retry-time *retryTime*
no tunnel mpls retry-time

- *retryTime* – interval from 1–60

Mode(s): Interface Configuration, Tunnel Profile Configuration

tunnel mpls vpn-id

- Description:** Associates a tunnel with a VPN. Specify the same values for the VPN ID that you specified when you associated a VR with the VPN ID. The **no** version removes the VPN ID from the virtual router.
- Syntax:** [no] tunnel mpls vpn-id oui *ouiNumber* index *ipAddress*
- *ouiNumber* – identifies the OUI portion of the VPN ID, ranges from 0–16777215
 - *ipAddress* – IP address that identifies the index portion of the VPN ID
- Mode(s):** Interface Configuration, Tunnel Profile Configuration

tunnel mtu

- Description:** Configures the maximum transmission unit size for the particular tunnel. The **no** version restores the default value: 1024 for DVMRP and GRE tunnels and 1440 for IPSec tunnels.
- Syntax:** tunnel mtu *mtuSize*
no tunnel mtu
- *mtuSize* – packet size allowed for transmission through the tunnel in the range: 1024–0240 bytes for DVMRP and GRE tunnels and 160–10240 for IPSec tunnels
- Mode(s):** Interface Configuration

tunnel password

- Description:** Configures a password for the L2TP tunnel. The **no** version removes the password.
- Syntax:** tunnel password *tunnelPassword*
no tunnel password
- *tunnelPassword* – password used for challenge response to the tunnel peer. In the domain map, it is used only by the LAC.
- Mode(s):** L2TP Destination Profile Host Configuration

tunnel peer-identity

- Description:** Specifies the peer identity of the IPsec tunnel. The **no** version removes the peer endpoint.
- Syntax:** tunnel peer-identity { address *ipAddress* | subnet *ipAddress subnetMask* | range *ipAddressLow ipAddressHigh* }
- no tunnel peer-identity
- address – specifies an IP address as the peer identity
 - subnet – specifies a subnet as the peer identity
 - range – specifies a range of IP addresses as the peer identity
- Mode(s):** Interface Configuration

tunnel pfs group

- Description:** Configures perfect forward secrecy for the IPsec tunnel by assigning a Diffie-Hellman prime modulus group. The **no** version removes PFS from this tunnel.
- Syntax:** tunnel pfs group { 1 | 2 | 5 }
- no tunnel pfs group
- 1 – 768-bit Diffie-Hellman prime modulus group
 - 2 – 1024-bit Diffie-Hellman prime modulus group
 - 5 – 1536-bit Diffie-Hellman prime modulus group
- Mode(s):** Interface Configuration

tunnel session-key-inbound

- Description:** Specifies the encryption and authentication algorithm set and session keys for manual inbound SAs. The **no** version removes the keys.
- Syntax:** tunnel session-key-inbound *inSaAlgorithms* { *encryptKey authKey* | *authKey* }
- no tunnel session-key-inbound
- *inSaAlgorithms* – algorithms to use for manual inbound SAs; use the online Help to see a list of available algorithms
 - *encryptKey* – encryption key; up to 48 characters
 - *authKey* – authentication key; up to 48 characters
- Mode(s):** Interface Configuration

tunnel session-key-outbound

- Description:** Specifies the encryption and authentication algorithm set, SPI, and session keys for manual outbound SAs. The **no** version removes the keys.
- Syntax:** tunnel session-key-outbound *outSAalgorithms spi { encryptKey authKey | encryptKey | authKey }*
no tunnel session-key-outbound
- *outSAalgorithms* – algorithms to use for manual outbound SAs; use the online Help to see a list of available algorithms
 - *spi* – number that uniquely identifies an SA; the range is 256 to 4294967295 (0xFFFFFFFF)
 - *encryptKey* – encryption key; up to 48 characters
 - *authKey* – authentication key; up to 48 characters
- Mode(s):** Interface Configuration

tunnel signaling

- Description:** Sets the signaling protocol used to negotiate security parameters and keys. The **no** version restores the default, isakmp.
- Syntax:** tunnel signaling { isakmp | manual }
no tunnel signaling
- *isakmp* – uses ISAKMP/IKE to negotiate parameters
 - *manual* – specifies that security parameters are configured manually
- Mode(s):** Interface Configuration

tunnel source

Description: Configures the source for a DVMRP, GRE, or IPSec tunnel. The **no** version deletes the tunnel source.

Syntax: For DVMRP and GRE tunnels:

```
tunnel source { ipAddress | interfaceType interfaceSpecifier }
```

```
no tunnel source
```

For IPSec tunnels:

```
tunnel source ipAddress
```

```
no tunnel source
```

- *ipAddress* – IP address of an existing interface that will serve as the tunnel's source
- *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
- *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*

Mode(s): Interface Configuration

tunnel transform-set

Description: Specifies a transform set that ISAKMP uses during SA negotiations on this tunnel. Transform sets used for manually configured tunnels can have only one transform. The **no** version removes the transform set from a tunnel.

Syntax: [no] tunnel transform-set *transformSetName*

- *transformSetName* – name of the transform set

Mode(s): Interface Configuration

type

Description: For RTR Configuration, configures an RTR operation. The **no** version removes the configured type from the operation and resets all configuration for an RTR index.



Note: You must configure the operation's type before you can configure any other characteristics of the operation.

For Domain Map Tunnel Configuration, specifies the type of tunnel as either L2TP or L2F. The **no** version restores the default value, L2TP.

Syntax: To configure an RTR operation:

```
[ no ] type rtrType protocol iplcmpEcho destination [ source-ipaddr srcAddr |
source ifType ifValue ]
```

- *rtrType* – one of the following types of operation:
 - › echo – performs end-to-end operation only
 - › pathEcho – discovers a path to the destination and echoes each device on the path
- *destination* – IP address or an IP hostname or domain name
- *srcAddr* – source IP address
- *ifType* – source's type of interface (for example, FastEthernet)
- *ifValue* – interface identifier; for example, FastEthernet 0/0

To specify the type of tunnel:

Syntax: type { l2tp | l2f }

no type

Mode(s): RTR Configuration, Domain Map Tunnel Configuration

udp-port

Description: Specifies the UDP port on the system where the RADIUS authentication or accounting servers reside. The system uses this port to communicate with the RADIUS servers. The **no** version restores the default value.

Syntax: udp-port *port*

no *udp-port*

- *port* – one of the following port numbers:
 - › 1812 (default for RADIUS authentication servers)
 - › 1813 (default for RADIUS accounting servers)

Mode(s): Radius Configuration

undebug ip bgp

Description: Turns off the display of information previously enabled with the **debug ip bgp** command. There is no **no** version.

Syntax: `undebug ip bgp [in | out] [peerAddress [peerAddressMask]]
[bgpLog] [import] [router routerName]
[filtering-router filteringRouterName] [accessClassName]
[route-map mapName]`

- `in` – displays information for inbound events
- `out` – displays information for outbound events
- `peerAddress` – IP address of BGP peer for which information is displayed
- `peerAddressMask` – network mask of BGP peer for which information is displayed
- `bgpLog` – BGP log of interest; one of the following options:
 - › `dampening` – BGP dampening event; route is suppressed or no longer suppressed by route-flap dampening
 - › `events` – BGP finite state machine events and transitions
 - › `keepalives` – BGP keepalive message events
 - › `updates` – BGP routing table update events
 - › `vpn4` – BGP VPNv4 NLRI events
- `import` – displays BGP import processing events; appears only if you specify the **vpn4** keyword
- `routerName` – name of the virtual router that owns the BGP router for which information is being displayed
- `filteringRouterName` – name of the virtual router that owns the access class and route map parameters
- `accessClassName` – name of an access list to filter output
- `mapName` – name of a route map to filter output

Mode(s): Privileged Exec

undebg ip mbgp

- Description:** Turns off the display of information previously enabled with the **debug ip mbgp** command. There is no **no** version.
- Syntax:** undebg ip mbgp [in | out] [*peerAddress* [*peerAddressMask*]] [*bgpLog*] [import] [router *routerName*] [filtering-router *filteringRouterName*] [*accessClassName*] [route-map *mapName*]
- in – displays information for inbound events
 - out – displays information for outbound events
 - *peerAddress* – IP address of BGP peer for which information is displayed
 - *peerAddressMask* – network mask of BGP peer for which information is displayed
 - *bgpLog* – BGP log of interest; one of the following options:
 - › dampening – BGP dampening event; route is suppressed or no longer suppressed by route-flap dampening
 - › events – BGP finite state machine events and transitions
 - › keepalives – BGP keepalive message events
 - › updates – BGP routing table update events
 - › vpv4 – BGP VPNv4 NLRI events
 - import – displays BGP import processing events; appears only if you specify the **vpnv4** keyword
 - *routerName* – name of the virtual router that owns the BGP router for which information is being displayed
 - *filteringRouterName* – name of the virtual router that owns the access class and route map parameters
 - *accessClassName* – name of an access list to filter output
 - *mapName* – name of a route map to filter output
- Mode(s):** Privileged Exec

undebg ip pim

- Description:** Turns off the display of information previously enabled with the **debug ip pim** command. There is no **no** version.
- Syntax:** undebg ip pim *pimLog*
- *pimLog* – PIM log of interest
- Mode(s):** Privileged Exec

undebug ip ospf

- Description:** Turns off the display of information for the selected variable. See **debug ip ospf** command for a complete list of the ospfLog variables. There is no **no** version.
- Syntax:** `undebug ip ospf ospfLog`
- *ospfLog* – OSPF log of interest; one of the following options:
 - › `adj` – OSPF adjacency events
 - › `elect-dr` – OSPF designated router election
 - › `events` – OSPF general events
 - › `lsa` – OSPF link state advertisements events
 - › `neighbor` – OSPF neighbor state machine
 - › `packets-rcvd` – OSPF packets received
 - › `packets-sent` – OSPF packets sent
 - › `route` – OSPF route events
 - › `spf` – all OSPF shortest path first calculation events
 - › `spf-ext` – OSPF shortest path first external route calculation events
 - › `spf-inter` – OSPF shortest path first interarea route calculation events
 - › `spf-intra` – OSPF shortest path first intra-area route calculation events
- Mode(s):** Privileged Exec

undebug ip rip

- Description:** Turns off the display of information previously enabled with the **debug ip rip** command. There is no **no** version.
- Syntax:** `undebug ip rip ripLog`
- *ripLog* – RIP log of interest; one of the following options:
 - › `events` – general RIP events, such as removing RIP from an interface or creating the RIP process
 - › `route` – events associated with two RIP routers exchanging routes
- Mode(s):** Privileged Exec

undebg isis

Description: Turns off the display of information for the selected variable. See the **debug isis** command for a complete list of the IS-IS log variables. There is no **no** version.

Syntax: undebg isis *isisLog*

- *isisLog* – IS-IS log of interest; one of the following options:
 - › adj-packets – IS-IS adjacency-related packets, such as hello packets sent and IS-IS received adjacencies going up and down
 - › mpls traffic-eng advertisements – MPLS traffic-engineering agent advertisements
 - › mpls traffic-eng agents – MPLS traffic-engineering agents
 - › snp-packets – IS-IS CSNPs/PSNPs
 - › spf-events – shortest path first events
 - › spf-statistics – SPF timing and statistic data
 - › spf-triggers – SPF triggering events
 - › update-packets – update-related packets

Mode(s): Privileged Exec

unframed

Description: Configures an unchannelized line for a CE1 interface. The **no** version switches the interface to the default crc4 framing mode and deletes channel group 1.

Syntax: [no] unframed

Mode(s): Controller Configuration

update-source

- Description:** Specifies the loopback interface whose local address is used as the source address for the OSPF, PIM, or RIP connection to a remote neighbor. The **no** version deletes the source address from the connection.
- Syntax:** For OSPF:
- [no] update-source loopback *interfaceSpecifier*
- *interfaceSpecifier* – integer in the range 1–4294967293 identifying the loopback interface
- For PIM
- [no] update-source *interfaceType interfaceSpecifier*
- *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
 - *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
- For RIP:
- [no] update-source *interfaceType interfaceSpecifier*
- *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
 - *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
- Mode(s):** Remote Neighbor Configuration

version

- Description:** Specifies the global RIP version. The **no** version reverts to the default value, RIP version 1. Use the `ip rip receive` and `ip rip send version` commands to specify the RIP version for a specific interface.
- Syntax:** version *versionNumber*
- no version
- *versionNumber* – RIP version number
- Mode(s):** Router Configuration

virtual-router

Description: When used from Domain Map Configuration mode, maps a virtual router to a user domain name. The **no** version deletes the virtual-router parameter, and the system defaults to the default virtual router.

When used from Privileged Exec or Global Configuration mode, creates a virtual router or accesses the context of a previously created virtual router or a VRF. The **no** version deletes the virtual router, and the system defaults to the default virtual router. Issuing a **no** version that specifies an existing VRF only displays the error message: “Cannot delete a VRF with this command”; you must use the **no ip vrf** command to remove a VRF.

Syntax: Mapping to a user domain name:

```
virtual-router vrName
```

```
no virtual-router [ vrName ]
```

Creating a virtual router or accessing a virtual router or VRF:

```
virtual-router vrName | :vrfName | vrName:vrfName
```

```
no virtual-router vrName
```

- *vrName* – name of the virtual router; a string of 1–15 alphanumeric characters
- *:vrfName* – specifies a VRF in the current VR context; a string of 1–32 alphanumeric characters
- *vrName:vrfName* – specifies a VRF in the context of a VR other than the current VR

Mode(s): Domain Map Configuration, Privileged Exec, Global Configuration

vlan id

Description: Specifies a VLAN ID. There is no **no** version.

Syntax: `vlan id IdValue [untagged]`

- *IdValue* – number in the range of 0–4095, which is unique within the Ethernet interface
- *untagged* – specifies that frames be sent untagged; only valid for VLAN ID 0

Mode(s): Interface Configuration

weight

- Description:** Sets the weighted round-robin weight of the scheduler node or queue. The **no** version sets the weight to the default value.
- Syntax:** `weight weightValue`
`no weight`
- *weightValue* – range 1-63; default is 8
- Mode(s):** Scheduler Profile Configuration

write memory

- Description:** Saves all outstanding (unsaved) configuration changes to nonvolatile storage; an exact alias of the **copy running-configuration startup-configuration** command. Available if the system is in either Automatic Commit mode or Manual Commit mode. If issued while in Automatic Commit mode, the CLI notifies you that the command is not necessary, but allows you to proceed. There is no **no** version.
- Syntax:** `write memory`
- Mode(s):** Privileged Exec