ZebOS-XP®
Network Platform

Version 1.4
Extended Performance

Unicast Routing Information Base
Command Reference

December 2015

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900
http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:
support@ipinfusion.com

Contents

Preface

This document describes the ZebOS-XP commands for the unicast Routing Information Base (RIB).

Audience

This document is intended for network administrators and other engineering professionals who configure and manage the ZebOS-XP unicast RIB.

Conventions

Table P-1 shows the conventions used in this guide.

Table P-1: Conventions

Convention	Description
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, functions, parameters, files, and directories

Contents

This document contains these chapters and appendices:

- Chapter 1, *Command Line Interface*
- Chapter 2, *Unicast RIB Commands*

Related Documents

The following guides are related to this document:

- *Unicast Configuration Guide*
- *Unicast Routing Information Base Developer Guide*
- *Installation Guide*

Note: All ZebOS-XP technical manuals are available to licensed customers at http://www.ipinfusion.com/support/document_list.

Support

For support-related questions, contact support@ipinfusion.com.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

CHAPTER 1 Command Line Interface

This chapter introduces the ZebOS-XP Command Line Interface (CLI) and how to use its features.

Overview

You use the CLI to configure, monitor, and maintain ZebOS-XP devices. The CLI is text-based and each command is usually associated with a specific task.

You can give the commands described in this manual locally from the console of a device running ZebOS-XP or remotely from a terminal emulator such as `putty` or `xterm`. You can also use the commands in scripts to automate configuration tasks.

Starting the Command Line Interface

You must start daemons as described in this section before you can use the CLI. The general steps are listed below. For details about the ZebOS-XP daemons, see the *Installation Guide.*

1. Start your terminal emulator and connect to the device or go to the console of the device running ZebOS-XP.
2. Connect to the directory where you installed the ZebOS-XP executables.
3. Start the Network Services Module (NSM).
   ```
   # ./nsm -d
   ```
4. Start the protocol module daemons that your organization uses, such as `mstpd`, `ospf6d`, or `ripd`.
   ```
   # ./mstpd -d
   ```
5. Start the Integrated Management Interface (IMI) daemon.
   ```
   # ./imi -d
   ```
6. Start the IMI shell.
   ```
   # ./imish
   ```
   Note: Your organization may use a ZebOS-XP build that does not include `imish`. If that is the case, you must connect to a port on which a protocol daemon is listening. For details, see the *Installation Guide.*

You can now begin using the CLI.

Command Line Interface Help

You access the CLI help by entering a full or partial command string and a question mark “?”. The CLI displays the command keywords or parameters along with a short description. For example, at the CLI command prompt, type:

```
> show ?
```

The CLI displays this keyword list with short descriptions for each keyword:

```
show ?
  application-priority            Application Priority
```

```
  arp                              Internet Protocol (IP)
  bfd                              Bidirectional Forwarding Detection (BFD)
  bgp                              Border Gateway Protocol (BGP)
  bi-lsp                           Bi-directional lsp status and configuration
  bridge                           Bridge group commands
  ce-vlan                          COS Preservation for Customer Edge VLAN
  class-map                        Class map entry
  cli                              Show CLI tree of current mode
  clns                             Connectionless-Mode Network Service (CLNS)
  control-adjacency                Control Adjacency status and configuration
  control-channel                  Control Channel status and configuration
  cspf                             CSPF Information
  customer                         Display Customer spanning-tree
  cvlan                            Display CVLAN information
  debugging                        Debugging functions (see also 'undebug')
  dot1x                            IEEE 802.1X Port-Based Access Control
  etherchannel                     LACP etherchannel
  ethernet                         Layer-2
...
```

If you type the ? in the middle of a keyword, the CLI displays help for that keyword only.

```
> show de?
debugging  Debugging functions (see also 'undebug')
```

If you type the ? in the middle of a keyword, but the incomplete keyword matches several other keywords, ZebOS-XP displays help for all matching keywords.

```
> show i? (CLI does not display the question mark).
interface  Interface status and configuration
ip         IP information
isis       ISIS information
```

Command Completion

The CLI can complete the spelling of a command or a parameter. Begin typing the command or parameter and then press the tab key. For example, at the CLI command prompt type `sh`:

```
> sh
```

Press the tab key. The CLI displays:

```
> show
```

If the spelling of a command or parameter is ambiguous, the CLI displays the choices that match the abbreviation. Type `show i` and press the tab key. The CLI displays:

```
> show i
  interface  ip         ipv6       isis
> show i
```

The CLI displays the `interface` and `ip` keywords. Type `n` to select `interface` and press the tab key. The CLI displays:

```
> show in
> show interface
```

Type ? and the CLI displays the list of parameters for the `show interface` command.

```
> show interface
  IFNAME  Interface name
  |       Output modifiers
```



```
>       Output redirection
<cr>
```

The CLI displays the only parameter associated with this command, the `IFNAME` parameter.

Command Abbreviations

The CLI accepts abbreviations that uniquely identify a keyword in commands. For example:

```
> sh in eth0
```

is an abbreviation for:

```
> show interface eth0
```

Command Line Errors

Any unknown spelling causes the CLI to display the error `Unrecognized command` in response to the ?. The CLI displays the command again as last entered.

```
> show dd?
% Unrecognized command
> show dd
```

When you press the Enter key after typing an invalid command, the CLI displays:

```
(config)#router ospf here
                           ^
% Invalid input detected at '^' marker.
```

where the ^ points to the first character in error in the command.

If a command is incomplete, the CLI displays the following message:

```
> show
% Incomplete command.
```

Some commands are too long for the display line and can wrap mid-parameter or mid-keyword, as shown below. This does *not* cause an error and the command performs as expected:

```
area 10.10.0.18 virtual-link 10.10.0.19 authent
ication-key 57393
```

Command Negation

Many commands have a `no` form that resets a feature to its default value or disables the feature. For example:

- The `ip address` command assigns an IPv4 address to an interface
- The `no ip address` command removes an IPv4 address from an interface

Syntax Conventions

Table 1-1 describes the conventions used to represent command syntax in this reference.

Table 1-1: Syntax conventions

Convention	Description	Example
`monospaced font`	Command strings entered on a command line	`show debugging rib`
`lowercase`	Keywords that you enter exactly as shown in the command syntax.	`show debugging rib`
`UPPERCASE`	See Variable Placeholders	`IFNAME`
`()`	Optional parameters, from which you must select one. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	`(A.B.C.D|<0-4294967295>)`
`()`	Optional parameters, from which you select one or none. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	`(A.B.C.D|<0-4294967295>|)`
`()`	Optional parameter which you can specify or omit. Do not enter the parentheses or vertical bar as part of the command.	`(IFNAME|)`
`{}`	Optional parameters, from which you must select one or more. Vertical bars delimit the selections. Do not enter the braces or vertical bars as part of the command.	`{intra-area <1-255>|inter-area <1-255>|external <1-255>}`
`[]`	Optional parameters, from which you select zero or more. Vertical bars delimit the selections. Do not enter the brackets or vertical bars as part of the command. A '?' before a parameter in square brackets limits that parameter to one occurrence in a command string.	`[<1-65535>|AA:NN|internet|local-AS|no-advertise|no-export]`
`.`	Repeatable parameter. The parameter that follows a period can be repeated more than once. Do not enter the period as part of the command.	`set as-path prepend .<1-65535>`

Variable Placeholders

Table 1-2 shows the tokens used in command syntax use to represent variables for which you supply a value.

Table 1-2: Variable placeholders

Token	Description
`WORD`	A contiguous text string (excluding spaces)
`LINE`	A text string, including spaces; no other parameters can follow this parameter
`IFNAME`	Interface name whose format varies depending on the platform; examples are: `eth0`, `Ethernet0`, `ethernet0`, `xe0`
`A.B.C.D`	IPv4 address
`A.B.C.D/M`	IPv4 address and mask/prefix
`X:X::X:X`	IPv6 address
`X:X::X:X/M`	IPv6 address and mask/prefix
`HH:MM:SS`	Time format
`AA:NN`	BGP community value
`XX:XX:XX:XX:XX:XX`	MAC address

`<1-5>`
`<1-65535>`
`<0-2147483647>`
`<0-4294967295>`
	Numeric range

Command Description Format

Table 1-3 explains the sections used to describe each command in this reference.

Table 1-3: Command descriptions

Section	Description
Command Name	The name of the command, followed by what the command does and when should it be used
Command Syntax	The syntax of the command
Parameters	Parameters and options for the command
Default	The state before the command is executed
Command Mode	The mode in which the command runs; see Command Modes
Example	An example of the command being executed

Keyboard Operations

Table 1-4 lists the operations you can perform from the keyboard.

Table 1-4: Keyboard operations

Key combination	Operation
Left arrow or Ctrl+b	Moves one character to the left. When a command extends beyond a single line, you can press left arrow or Ctrl+b repeatedly to scroll toward the beginning of the line, or you can press Ctrl+a to go directly to the beginning of the line.
Right arrow or Ctrl-f	Moves one character to the right. When a command extends beyond a single line, you can press right arrow or Ctrl+f repeatedly to scroll toward the end of the line, or you can press Ctrl+e to go directly to the end of the line.
Esc, b	Moves back one word
Esc, f	Moves forward one word
Ctrl+e	Moves to end of the line
Ctrl+a	Moves to the beginning of the line
Ctrl+u	Deletes the line
Ctrl+w	Deletes from the cursor to the previous whitespace
Alt+d	Deletes the current word
Ctrl+k	Deletes from the cursor to the end of line
Ctrl+y	Pastes text previously deleted with Ctrl+k, Alt+d, Ctrl+w, or Ctrl+u at the cursor

Table 1-4: Keyboard operations (Continued)

Key combination	Operation
Ctrl+t	Transposes the current character with the previous character
Ctrl+c	Ignores the current line and redisplays the command prompt
Ctrl+z	Ends configuration mode and returns to exec mode
Ctrl+l	Clears the screen
Up Arrow or Ctrl+p	Scroll backward through command history
Down Arrow or Ctrl+n	Scroll forward through command history

Show Command Modifiers

You can use two tokens to modify the output of a `show` command. Enter a question mark to display these tokens:

```
# show users ?
  | Output modifiers
  > Output redirection
```

You can type the | (vertical bar character) to use output modifiers. For example:

```
> show rsvp | ?
  begin     Begin with the line that matches
  exclude   Exclude lines that match
  include   Include lines that match
  redirect  Redirect output
```

Begin Modifier

The `begin` modifier displays the output beginning with the first line that contains the input string (everything typed after the `begin` keyword). For example:

```
# show run | begin eth1
...skipping
interface eth1
 ipv6 address fe80::204:75ff:fee6:5393/64
!
interface eth2
 ipv6 address fe80::20d:56ff:fe96:725a/64
!
line con 0
 login
!
end
```

You can specify a regular expression after the `begin` keyword, This example begins the output at a line with either “eth3” or “eth4”:

```
# show run | begin eth[3-4]

...skipping
interface eth3
```

```
 shutdown
!
interface eth4
 shutdown
!
interface svlan0.1
 no shutdown
!
route-map myroute permit 3
!
route-map mymap1 permit 10
!
route-map rmap1 permit 3
!
line con 0
 login
line vty 0 4
 login
!
end
```

Include Modifier

The `include` modifier includes only those lines of output that contain the input string. In the output below, all lines containing the word “input” are included:

```
# show interface eth1 | include input
    input packets 80434552, bytes 2147483647, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 1, missed 0
```

You can specify a regular expression after the `include` keyword. This examples includes all lines with “input” or “output”:

```
#show int eth0 | include (in|out)put
    input packets 597058, bytes 338081476, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 613147, bytes 126055987, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
```

Exclude Modifier

The `exclude` modifier excludes all lines of output that contain the input string. In the following output example, all lines containing the word “input” are excluded:

```
# show interface eth1 | exclude input
Interface eth1
  Scope: both
  Hardware is Ethernet, address is 0004.75e6.5393
  index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Administrative Group(s): None
  DSTE Bandwidth Constraint Mode is MAM
  inet6 fe80::204:75ff:fee6:5393/64
    output packets 4438, bytes 394940, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
```

You can specify a regular expression after the `exclude` keyword. This example excludes lines with “output” or “input”:


```
# show interface eth0 | exclude (in|out)put
Interface eth0
  Scope: both
  Hardware is Ethernet  Current HW addr: 001b.2139.6c4a
  Physical:001b.2139.6c4a  Logical:(not set)
  index 2 metric 1 mtu 1500 duplex-full arp ageing timeout 3000
  <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Bandwidth 100m
  DHCP client is disabled.
  inet 10.1.2.173/24 broadcast 10.1.2.255
  VRRP Master of :  VRRP is not configured on this interface.
  inet6 fe80::21b:21ff:fe39:6c4a/64
    collisions 0
```

Redirect Modifier

The `redirect` modifier writes the output into a file. The output is not displayed.

```
# show history | redirect /var/frame.txt
```

The output redirection token (>) does the same thing:

```
# show history >/var/frame.txt
```

Command Modes

Commands are grouped into modes arranged in a hierarchy. Each mode has its own set of commands. Table 1-5 lists the command modes common to all protocols.

Table 1-5: Common command modes

Name	Description
Executive mode	Also called *view* mode, this is the first mode to appear after you start the CLI. It is a base mode from where you can perform basic commands such as `show`, `exit`, `quit`, `help`, `list`, and `enable`.
Privileged executive mode	Also called *enable* mode, in this mode you can run additional basic commands such as `debug`, `write`, and `show`.
Configure mode	Also called *configure terminal* mode, in this mode you can run configuration commands and go into other modes such as interface, router, route map, key chain, and address family.
Interface mode	In this mode you can configure protocol-specific settings for a particular interface. Any setting you configure in this mode overrides a setting configured in router mode.
Router mode	This mode is used to configure router-specific settings for a protocol such as RIP or OSPF.

Command Mode Tree

The diagram below shows the common command mode hierarchy.

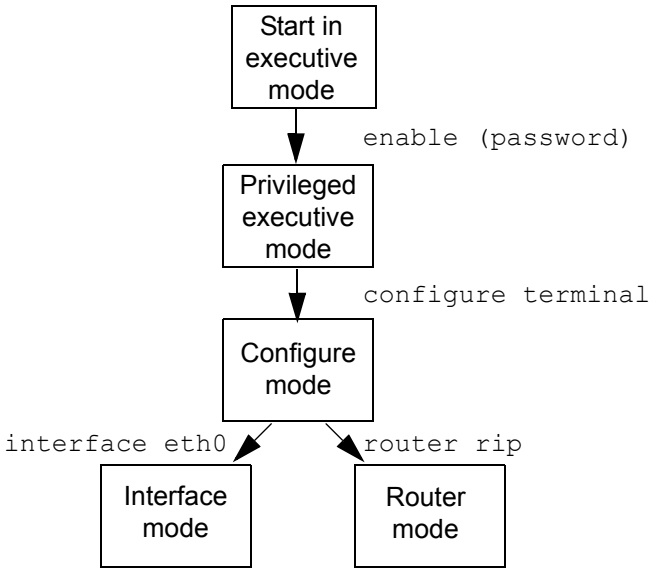

Figure 1-1: Common command modes

To change modes:

1. Enter privileged executive mode by entering `enable` in Executive mode.
2. Enter configure mode by entering `configure terminal` in Privileged Executive mode.

The example below shows starting `imish` and then moving from executive mode to privileged executive mode to configure mode and finally to router mode:

```
# ./imish
> enable mypassword
# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
(config)# router rip
(config-router)#
```

Note: Each protocol can have modes in addition to the common command modes. See the command reference for the respective protocol for details.

Debug Command

Whether the settings you make for a `debug` command persist between sessions depends on the mode where you make the settings:

- When you make settings for a `debug` command in executive mode, the configuration is valid for the current session only and is not saved in the `ZebOS.conf` file.
- When you make settings for a `debug` command in configuration mode, the configuration is retained and saved in `ZebOS.conf` and used even after the session restarts.

CHAPTER 2 Unicast RIB Commands

This chapter describes the following unicast RIB commands:

clear ip route kernel

Use this command to clear stale IPv4 routes from the RIB (Routing Information Base) and FIB (Forwarding Information Base).

Command Syntax

```
clear ip route kernel
```

Parameters

None

Command Mode

Privileged Exec mode

Example

```
#clear ip route kernel
```

clear ipv6 route kernel

Use this command to clear stale IPv6 routes from the RIB (Routing Information Base) and FIB (forwarding Information Base).

Command Syntax

```
clear ipv6 route kernel
```

Parameters

None

Command Mode

Privileged Exec mode

Example

```
#clear ipv6 route kernel
```

debug rib

Use this command to debug the `ribd` process.

Use the `no` form of this command or the `undebug` command to stop debugging.

Note: High-availability and kernel events are not supported in ZebIC releases.

Command Syntax

```
debug rib (all|)
debug rib events
debug rib packet (recv|send|) (detail|)
debug rib kernel
debug rib ha
debug rib ha all
debug rib nsm
debug rib bfd
debug ip routing (add-route|delete-route|mod-route|
debug ipv6 routing (add-route|delete-route|mod-route|)
debug rib hal events
no debug all
no debug rib (all|)
no debug all rib
no debug rib kernel
no debug rib events
no debug rib packet (recv|send|) (detail|)
no debug rib kernel
no debug rib ha
no debug rib ha all
no debug rib nsm
no debug rib bfd
no debug ip routing (add-route|delete-route|mod-route|)
no debug ipv6 routing (add-route|delete-route|mod-route|)
no debug rib hal events
undebug all
undebug rib (all|)
undebug all rib
undebug rib events
undebug rib packet (recv|send|) (detail|)
undebug rib kernel
```

```
undebug rib ha
undebug rib ha all
undebug rib nsm
undebug rib bfd
undebug rib hal events
```

Parameters

`all`	All debugging functions
`events`	Events
`packet`	Packet events
`recv`	Received packets
`send`	Sent packets
`detail`	Detailed information
`kernel`	Kernel events
`ha`	High-availability events
`all`	All debugging functions
`nsm`	NSM events
`bfd`	BFD (Bidirectional Forwarding Detection) events
`ip routing`	IPv4 routing events
`ipv6 routing`	IPv6 routing events
`add-route`	Add route events
`delete-route`	Delete route events
`mod-route`	Modify route events
`hal events`	HAL (Hardware Abstraction Layer) events

Command Mode

Privileged Exec mode

Example

```
#debug rib events
```

description

Use this command to assign a description to a Virtual Router instance.

Use the `no` parameter to remove a description.

Command Syntax

```
description LINE
no description
```

Parameters

`LINE`	Virtual Router description

Command Mode

VR mode

Examples

```
#configure terminal
(config)#virtual-router VR1
(config-vr)#description VR1 has been created for CLI testing
(config-vr)#exit

(config)#virtual-router VR1
(config-vr)#no description
(config-vr)#exit
```

fib retain

Use this command to set the retention time for stale routes in the Forwarding Information Base (FIB) when `ribd` restarts. The `ribd` process reads the FIB and treats previously self-installed routes as stale.

You can display stale routes by running the `show ip route database` command. All routes preceded by the symbol `p` are stale routes. When protocol modules restart, `ribd` overrides these stale routes with routes updated by the protocol modules.

Table 2-1 show the behavior of routes when `ribd` stops.

Table 2-1: FIB retention

Command	Behavior
`fib retain`	Does not clear routes from the FIB and retains stale routes for 60 seconds when restarted.
`fib retain forever`	Does not clear routes and retains stale routes forever.
`fib retain time <1-65535>`	Does not clear routes and retains stale routes for the specified seconds.
`no fib retain` (default)	Cleans up routes in the FIB, but retains stale routes for 60 seconds when restarted.

You can remove stale routes at any time with the clear ip route kernel command.

Use the `no` form of this command to revert to default; that is, do not retain routes in the FIB when `ribd` stops.

Command Syntax

```
fib retain (forever|time <1-65535>|)
no fib retain (forever|time <1-65535>|)
```

Parameters

`forever`	Retain FIB forever
`time`	Retain FIB for a time after `ribd` restarts
`<1-65535>`	Retention time in seconds; if you omit this value, the default is 60 seconds

Default

Routes are cleared from the FIB when `ribd` stops. However, when `ribd` restarts, stale routes are retained for 60 seconds.

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#fib retain time 180
```

ip route

Use this command to create an IPv4 static route.

Use the `no` form of this command to delete a static route.

Command Syntax

```
ip route A.B.C.D/M (A.B.C.D|IFNAME)
ip route A.B.C.D/M A.B.C.D IFNAME
ip route A.B.C.D A.B.C.D (A.B.C.D|IFNAME)
ip route A.B.C.D A.B.C.D A.B.C.D IFNAME
ip route A.B.C.D/M (A.B.C.D|IFNAME){<1-255>|tag <0-4294967295>|description WORD}
ip route A.B.C.D/M A.B.C.D IFNAME {<1-255>|tag <0-4294967295>|description WORD}
ip route A.B.C.D A.B.C.D (A.B.C.D|IFNAME){<1-255>|tag <0-4294967295>|description
  WORD}
ip route A.B.C.D A.B.C.D A.B.C.D IFNAME {<1-255>|tag <0-4294967295>|description
  WORD}
ip route vrf NAME A.B.C.D/M IFNAME
ip route vrf NAME A.B.C.D/M A.B.C.D IFNAME
ip route vrf NAME A.B.C.D/M IFNAME {<1-255>|tag <0-4294967295>|description WORD}
ip route vrf NAME A.B.C.D/M A.B.C.D IFNAME {tag <0-4294967295>|description WORD}
no ip route A.B.C.D/M (A.B.C.D|IFNAME|)
no ip route A.B.C.D/M A.B.C.D IFNAME
no ip route A.B.C.D A.B.C.D (A.B.C.D|IFNAME)
no ip route A.B.C.D A.B.C.D A.B.C.D IFNAME
no ip route A.B.C.D/M (A.B.C.D|IFNAME){<1-255>|tag <0-4294967295>|description WORD}
no ip route A.B.C.D/M A.B.C.D IFNAME {<1-255>|tag <0-4294967295>|description WORD}
no ip route A.B.C.D A.B.C.D (A.B.C.D|IFNAME){<1-255>|tag <0-4294967295>|description
  WORD}
no ip route A.B.C.D A.B.C.D A.B.C.D IFNAME {<1-255>|tag <0-4294967295>|description
  WORD}
no ip route vrf NAME A.B.C.D/M IFNAME
no ip route vrf NAME A.B.C.D/M A.B.C.D IFNAME
no ip route vrf NAME A.B.C.D/M IFNAME {tag <0-4294967295>|description WORD}
no ip route vrf NAME A.B.C.D/M A.B.C.D IFNAME {tag <0-4294967295>|description WORD}
```

Parameters

`A.B.C.D/M`	Subnet: IP destination prefix and a mask length
`A.B.C.D A.B.C.D`	Subnet: IP destination address and mask
`A.B.C.D`	Gateway nexthop IPv4 address

`<1-255>`	Administrative distance
`IFNAME`	Gateway nexthop interface name
`description`	Description of the static route
`tag`	Tag used as a “match” value to control redistribution via route maps
`<0-4294967295>`	Tag value
`vrf`	VRF (Virtual Routing and Forwarding) instance
`NAME`	VRF name

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip route 192.168.3.0 255.255.255.0 2.2.2.2 128
(config)#ip route 1.1.1.0/24 eth0 32
(config)#ip route vrf new 1.1.1.1/1 1.1.1.1 eth1 description new tag 1
```

ip vrf

This command creates a VRF (Virtual Routing and Forwarding) RIB (Routing Information Base), assigns a VRF identifier, and switches to VRF mode.

Use the `no` parameter with command to remove a VRF RIB.

Command Syntax

```
ip vrf WORD
no ip vrf WORD
```

Parameter

`WORD`	VRF identifier

Command Mode

Configure mode

Example

```
#configure terminal
(config)#ip vrf IPI
(config-vrf)#
```

ip vrf isid

Use this command to create a VRF (Virtual Routing and Forwarding) instance associated with an I-SID (service instance identifier) that needs to advertise its routes over an SPB network.

Use the `no` form of this command to remove a VRF.

Note: This command is not supported for ZebIC releases.

Command syntax

```
ip vrf WORD isid <1-16777214>
no ip vrf WORD
```

Parameters

`WORD`	VRF instance name
`<1-16777214>`	Service instance identifier

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip vrf vpn2 isid 2
```

ipv6 route

Use this command to create an IPv6 static route for a subnet.

Use the `no` form of this command to delete a static route.

Command Syntax

```
ipv6 route X:X::X:X/M (X:X::X:X|IFNAME)
ipv6 route X:X::X:X/M X:X::X:X IFNAME
ipv6 route vrf NAME X:X::X:X/M X:X::X:X IFNAME
ipv6 route X:X::X:X/M (X:X::X:X|IFNAME) <1-255>
ipv6 route X:X::X:X/M X:X::X:X IFNAME <1-255>
no ipv6 route X:X::X:X/M
no ipv6 route X:X::X:X/M (X:X::X:X|IFNAME)
no ipv6 route X:X::X:X/M X:X::X:X IFNAME
no ipv6 route vrf NAME X:X::X:X/M X:X::X:X IFNAME
no ipv6 route X:X::X:X/M (X:X::X:X|IFNAME) <1-255>
no ipv6 route X:X::X:X/M X:X::X:X IFNAME <1-255>
```

Parameters

`X:X::X:X/M`	Subnet: IPv6 destination prefix and mask length <0-128>
`X:X::X:X`	Gateway nexthop IPv6 address
`IFNAME`	Gateway nexthop interface name
`vrf`	VRF (Virtual Routing and Forwarding) instance
`NAME`	VRF name
`<1-255>`	Administrative distance

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ipv6 route 3ffe:506::1/128 eth1 32
```

maximum-paths

Use this command to set the maximum number of paths to install in the FIB (Forwarding Information Base) for the ECMP (Equal-Cost MultiPath) feature.

Use the `no` parameter with this command to revert to default.

Command Syntax

```
maximum-paths <1-64>
no maximum-paths <1-64>
no maximum-paths
```

Parameter

`<1-64>`	Maximum number of paths to install in the FIB

Default

By default, the maximum number of paths is 8.

Command Mode

Configure mode

Example

```
#configure terminal
(config)#maximum-paths 5
```

max-fib-routes

Use this command to set the maximum number of FIB (Forwarding Information Base) routes excluding kernel, connected, and static routes.

Use the `no` parameter to remove this configuration.

Command Syntax

```
max-fib-routes <1-4294967294>
no max-fib-routes
```

Parameters

`<1-4294967294>` Maximum number of FIB routes, excluding kernel, connected, and static routes

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#max-fib-routes 12345

(config)#no max-fib-routes
```

max-static-routes

Use this command to set the maximum number of static routes.

Use the `no` parameter to disable this command.

Command Syntax

```
max-static-routes <1-4294967294>
no max-static-routes
```

Parameters

`<1-4294967294>` Maximum number of static routes

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#max-static-routes 123

(config)#no max-static-routes
```

show debugging rib

Use this command to display debug settings.

Command Syntax

```
show debugging rib
```

Parameters

None

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#show debugging rib
```

snmp restart rib

Use this command to restart SNMP in Routing Information Base (RIB)

Command Syntax

```
snmp restart rib
```

Parameters

None

Command Mode

Configure mode

Examples

```
#snmp restart rib
```

Index