

Cisco IOS Quick Reference

Cisco IOS Quick Reference.....	1	Show commands.....	3
Router modes and prompts.....	1	ip.....	3
Moving through modes.....	1	Filtering.....	4
Global configuration cmds.....	1	Debug commands.....	4
Line configuration.....	2	Appendix: Abbreviations and acronyms.....	4
Privileged mode commands.....	2	Appendix: powers of 2.....	4
Interface configuration.....	2	Appendix: Cidr and block.....	4
Subinterface configuration.....	3	Appendix: subnetting.....	4
Standard named ACL configuration.....	3	steps.....	4
VLAN.....	3	Example.....	4
Keys combinations.....	3	Appendix: variable length subnetting.....	4

Conventions

(explanation)	text in parentheses is an explanation; don't type this
enable	command; type it as written
<i>name</i>	user argument to a command; type the value you care, the indicated value is an example
<u>option</u>	keyword to a command; type an accepted value; if in [] it's optional ; if in { } it's mandatory ; a list of the form ..., it's the list of acceptable keywords
n	footer note: more explanation; don't type it
R> R# R(config)# ...	prompt of Cisco IOS router (or switch) shell; R (or S) is hostname; you don't type these
ipv6 ...	command about IPv6 configuration
to check	this command needs more investigations

Intermediate device (router, switch, ...) has different modes of operation. Every mode has its set of commands and it has a different command prompt. Below, when you consult a command, watch at the relative prompt, on the left side, to know in what mode it is available.

Below, R in prompt may indicate the command is about a router only, or it may indicate it is about a generic intermediate device. S in prompt indicates the command is about a switch only.

Router modes

R# disable

Prompts

- R> (user-mode)
- R# (privileged-mode)
- R(config)# (global configuration)
- R(config-if)# (configure interface)
- R(config-line)# (configure line)
- R(config-subif)# (configure subinterface)
- R(config-std-nacl)# (configure standard named ACL)

Moving through modes

user	privileged	global conf.	config line/if
>	#	(config)#	(config-line)#
x -- enable -->	x-- configure terminal	->x--- line [] -->x	
	x<-----	/- ctrl+z V end -x (to #)	
x <- exit / CR -x <-----	exit -----	x<-- exit	-----x
x <- disable -- x			

Ctrl+Z (or end) goes to *privileged-mode* either from *config* and from *config-line*

```
R> enable
R# ?1
```

1 help about commands

Procedure: how to start

Initial configuration

```
R> enable
R# configure terminal
R(config)# no ip domain-lookup
R(config)# hostname R1
S(config)# ip default-gateway ip-addr
R1(config)# ip domain-name ldfa.it
R1(config)# ipv6 unicast-routing
```

Restrict access to privileged mode

```
R1(config)# enable secret pass1234*
```

Restrict access to user mode

```
R1(config)# service password-encryption
```

```
R1(config)# login block-for 120 attempts 3 within 60
R1(config)# security password min-length 8
```

```
(enable ssh)
R1# show ip ssh
```

```
R1(config)# crypto key generate rsa general-keys
modulus 1024
```

```
R1(config)# [no] username luciano privilege 15 secret
pass1234*
```

```
R1(config)# line console 0
R1(config-line)# password pass4321*
R1(config-line)# login
```

```
R1(config-line)# line vty 0 15
R1(config-line)# password pass4321*
R1(config-line)# login local
R1(config-line)# transport input ssh
S(config-line)# exec timeout 10
```

```
R1(config-line)# line aux 0
R1(config-line)# password pass4321*
R1(config-line)# login local
R1(config-line)# transport input ssh
S(config-line)# exec timeout 10
```

```
R1(config)# banner motd delimiter message delimiter
```

Interface configuration

(reiterate for every interface to configure)

```
R1(config)# interface {vlan 1 | gigabitethernet 0/0 |
fastethernet 0/0 | serial 0/0/0 | loopback}
R1(config-if)# ip address ip-addr subnet-mask
R1(config-if)# ipv6 enable
R1(config-if)# ipv6 address ipv6-addr/prefix-len
R1(config-if)# ipv6 address FE80::1 link-local
R1(config-if)# description interface_description
R1(config-if)# no shutdown
```

to privileged mode

```
R1(config-if)# end
```

verify configuration

```
R1# show running-config
R1# show ip interface [brief]
R1# show ipv6 interface [brief]
R1# ping ip-addr
R1# ping ipv6 ipv6-addr
R1# show ip route [static | dynamic]
R1# show ipv6 route [static | address]
R1# show interfaces
R1# show interface interface
R1# traceroute ip-addr
```

save configuration

```
R1# copy running-config startup-config
```

to user mode

```
R1# disable
```

Global configuration cmds

```
R(config)# [no] access-list acl_id {deny | permit |
remark} source [wildcard] [log]
R(config)# [no] access-list acl-id {deny | permit |
remark} host source [log]
R(config)# banner motd delimiter message delimiter 2
S(config)# boot system
flash:/c2960-lanbasek9-mz.150-2.SE/c2960-lanbase
k9-mz.150-2.SE.bin 3
R(config)# crypto key generate rsa general-keys
modulus 1024 4
R(config)# crypto key zeroize rsa 5
R(config)# no cdp run 6
R(config)# enable {password | secret} password 7
R(config)# [no8] hostname host-name 9
R(config)# ip access-list {standard | extended} name 10
R(config)# no ip domain-lookup 11
R(config)# interface gigabitethernet 0/0/10 12
R(config)# interface range gigabitethernet 0/0-15 13
R(config)# [no] ip access-group acl-id {in | out} 14
S(config)# ip default-gateway ip-addr 15
R(config)# ip domain-name domain_name 16
R(config)# ip route net_address subnet_mask
{next_hop_address | interface} [distance] 17
R(config)# ipv6 route net_ip_address/prefix_len
{next_hop_address | interface} 18
```

- 2 delimiter is a single character, usually #
- 3 set the boot env.variable to: storage device (flash), path to dir in file system (:/c2960...), filename of IOS (/c2960-...bin)
- 4 generate key needed by SSH. IOS filename must contain the string "k9"
- 5 erase all SSH configurations and it stops SSH server
- 6 disable globally cisco discovery protocol
- 7 privileged mode password stored in plain text / hashed form
- 8 **no hostname**: set hostname to factory default
- 9 **host-name**: max len 64 chars, only {letters | digits | dashes}, start with letter
- 10 create named ACL, then go to standard named ACL mode
- 11 disable DNS lookup, beware: it fights with next command
- 12 select interface: {*vlan 1 | gigabitethernet 0/0 | fastethernet 0/0 | serial 0/0/0 | loopback*}. If it uses a *g0/0.10* schema, it is a subinterface definition; usually the subinterface number (here: 10) is the same of the vlan id.
- 13 select a group of interfaces to modify globally: {*vlan min-max | gigabitethernet 0/min-max | fastethernet 0/min-max* }
- 14 activate ACL on the interface
- 15 **no on routers**, unless routing is disabled
- 16 needed by SSH
- 17 configure an IPv4 static route; to configure a gateway of last resort, *net_address* and *subnet_mask* should be: 0.0.0.0 0.0.0.0

```
R(config)# ipv6 unicast-routing19
R(config)# ip ssh version 220
R(config)# line {console 0 | vty 0 15 | aux 0}21
R(config)# login block-for 120 attempts 3 within 6022
R(config)# router ?23
R(config)# security password min-length 8
R(config)# service password-encryption24
R(config)# username user_name secret password25
S1(config)# [no] vlan vlan_id26
```

Line configuration

```
(select line type, then:)
R(config-line)# password password
R(config-line)# login [local27]
R(config-line)# transport input ssh28
R(config-line)# exec timeout 1029
R(config-line)# exit
```

Privileged mode commands

```
R# clear arp-cache30
S# clear mac address-table dynamic31
R# clock set hh:mm:ss MMM dd YYYY
S# delete flash:vlan.dat32
R# erase startup.config
R# ping ip-addr
R# ping ipv6 ipv6-addr
R# reload33
R# terminal history size 200
R# terminal [no] monitor34
R# terminal [no35] length 036
R# traceroute ip-addr
```

- 18 configure an IPv6 static route; to configure a gateway of last resort, *net_ip_address/prefix_len* must be *::/0*
- 19 enable IPv6 routing; without this *ipv6* cmds are usefulness
- 20 force to use SSH ver.2 (ver.1 has security flaws)
- 21 select line
- 22 block login for 120 secs if fails 3 times in 60 secs
- 23 list supported IPv4 dynamic routing protocols
- 24 perform password encryption service
- 25 create user for SSH
- 26 (revert to the indicated *vlan_id*) create/select a VLAN. *vlan_id* could be a range, as: 1, 2, 3, or: 1-3.
- 27 in case of SSH
- 28 to use SSH
- 29 **switch only:** disconnect idle users after 10 minutes
- 30 in windows: `arp -d`
- 31 reset table port/mac addresses
- 32 remove all VLANs (after this. reload the switch)
- 33 reboots the router, loads startup-config
- 34 enable/disable log output to terminal vty
- 35 `terminal no length` restore default behaviour
- 36 0 length: doesn't pause output; not functioning in packet tracer

Interface configuration

```
(select interface, then:)
R(config-if)# description interface_description37
R(config-if)# ip access-group {acl-id | acl-name} {in | out}
R(config-if)# ip address ip-addr subnet-mask
R(config-if)# ipv6 enable38
R(config-if)# ipv6 address ipv6-addr/prefix-len
R(config-if)# ipv6 address FE80::1 link-local39
R(config-if)# no cdp enable40
R(config-if)# clock rate 12800041
S(config-if)# duplex {auto | half | full}
S(config-if)# speed {auto | 100 | ...}
S(config-if)# mdix {auto | ...}42
S(config-if)# mls qos trust [cos | device cisco-phone | dscp | ip-precedence]43
R(config-if)# no shutdown
S(config-if)# [no] switchport access vlan vlan_id44
S(config-if)# switchport mode {access | trunk}45
S(config-if)# switchport port-security46
S(config-if)# switchport port-security mac-address allowed_address47
S(config-if)# [no] switchport port-security mac-address sticky48
S(config-if)# switchport port-security violation {protect | restrict | shutdown}
S(config-if)# [no] switchport trunk allowed vlan vlan-list49
S(config-if)# [no] switchport trunk native vlan vlan-id50
S(config-if)# switchport voice vlan vlan-id51
R(config-if)# exit
```

- 37 max description length: 240 characters
- 38 create link local IPv6 address even without IPv6 global unicast address has been assigned
- 39 IPv6 link local address is in FE80::/10, this is an example. Memo: ::1 is the IPv6 loopback address.
- 40 disable cdp on this interface
- 41 this only on serial interface **to DCE plug** of the cable
- 42 this requires *duplex* and *speed* set to *auto*
- 43 set the trusted state and indicate which fields are used to classify traffic. necessary for voice ip traffic
- 44 (remove from VLAN; in this case no *vlan_id*) assign interface to VLAN, you can assign it directly to another VLAN
- 45 *mode access*: port used to link a host, it accepts only one VLAN. *mode trunk*, in this case more VLANs can traverse it, used to link another switch or a router; default: allowed all VLANs, native is vlan 1.
- 46 enable secure MAC address
- 47 set static secure MAC address
- 48 set (or unset) sticky secure MAC address
- 49 set the allowed VLANs in trunk. *no* (without *vlan-list*): all VLANs allowed
- 50 set native VLAN. *no* (without *vlan-id*): sets native vlan back to vlan 1
- 51 VLAN used to carry voice

Subinterface configuration

(select subinterface, then:)

```
R(config-subif)# encapsulation dot1q vlan-id 52
```

(then, assign ip address as interface; when done, return to interface and activate it⁵³)

Standard named ACL configuration

(create named CAL, then:)

```
R(config-std-nacl)# {permit | deny | remark} source  
[wilrdcard] [log]
```

VLAN

(select vlan, then:)

```
S(config-vlan)# name vlan_name 54
```

```
S(config-vlan)# exit
```

Keys combinations

command completion: **tab**

escape sequence: **Ctrl+Shift+6**

almost as end: **Ctrl+Z** ⁵⁵

previous cmd in history: **Ctrl+P** or **Up arrow**

next cmd in history: **Ctrl+N** or **Down arrow**

Show commands

```
R# show access-lists
```

```
R# show cdp neighbors detail
```

```
R# show clock
```

```
S# show controllers ethernet-controller fa 0/1 phy 56
```

```
S# show flash
```

```
R# show history
```

```
R# show interface interface-id status
```

```
R# show interfaces {interface-id | vlan vlan_id |  
switchport} 57
```

```
S# show mac address-table 58
```

```
S# show port-security address [interface-id] 59
```

```
S# show port-security interface [interface-id]
```

52 define encapsulation type for subinterface on trunk. it is possible append the keyword native to set the native vlan.

53 interface activate all its subinterfaces. It is possible to (de)activate every single subinterface with the [no] shutdown command.

54 create VLAN

55 exec current command if existent; the end command doesn't

56 show physical configuration of a single interface

57 statistics for all interfaces. *vlan* and *switchport* is about switch layer 2 only

58 show the port/mac addresses table of the switch. another form could be show mac-address-table

59 display all secure MAC addresses configured on all switch interfaces, or on a specified interface

```
R# show protocols
```

```
R# show privilege 60
```

```
R# show running-config [interface gigabitethernet 0/0] 61
```

```
R# show ssh 62
```

```
R# show version
```

```
S# show vlan {brief | id vlan_id | name vlan_name |  
summary}
```

ip

```
R# show ip arp 63
```

```
R# show ip interface [brief] 64
```

```
R# show ipv6 interface [brief]
```

```
S# show ip [interface-id]
```

```
R# show ip route [static | dynamic] 65
```

```
R# show ipv6 route [static 66 | network]
```

```
S# show ip ssh 67
```

Filtering

```
R# show ... | {section | include | exclude | begin} a_filter
```

Debug commands

```
R# [no] debug ip icmp
```

```
R# [no] debug all (not recommended)
```

```
R# [no] debug ip packet (not recommended)
```

```
R# undebug all 68
```

Appendix: Abbreviations and acronyms

ACL Access Contrl List

addr. address

b.cast broadcast address

b.sz block size

CDP Cisco Discovery Protocol

DCE Data Circuit-terminating Equipment

h.m. how many

if Interface

IOS cisco Internetworking Operating System

IP Internetwork protocol

LAN Local Area Network

MAC Media Access Control

n. number

s.mask subnet mask

SSH Secure Shell

SVI Switch Virtual Interface

VLAN Virtual LAN

60 user is 1, privileged is 15

61 not functioning in packet tracer

62 check SSH connections to device

63 On windows: arp -a

64 IPv4 interfaces status; if brief is present, it's in table format with more infos, as IP addresses ...

65 shows routing table

66 not functioning in packet tracer

67 to verify if SSH is supported, otherwise the command isn't recognized

68 switch off all debug commands

Appendix: powers of 2

$2^1 = 2$	$2^8 = 256$
$2^2 = 4$	$2^9 = 512$
$2^3 = 8$	$2^{10} = 1024$
$2^4 = 16$	$2^{11} = 2048$
$2^5 = 32$	$2^{12} = 4096$
$2^6 = 64$	$2^{13} = 8192$
$2^7 = 128$	$2^{14} = 16384$

Appendix: Cidr and block

prefix	bin (x,y)		mask	nets (2^x)	blk size (2^y)	hosts (2^y-2)
/24	0000 0000		0	1	256	254
/25	1000 0000	+128	128	2	128	126
/26	1100 0000	+64	192	4	64	62
/27	1110 0000	+32	224	8	32	30
/28	1111 0000	+16	240	16	16	14
/29	1111 1000	+8	248	32	8	6
/30	1111 1100	+4	252	64	4	2
/31	1111 1110	+2	254	128	2	0
/32	1111 1111	+1	255	256	1	-1

Appendix: subnetting

steps

	Question	How calculate
1	n.of subnets	2^x (x=n.of masked bits)
2	h.m.hosts in subnet	$2^y - 2$ (y=n.of unmasked bits)
3	list subnets	<ul style="list-style-type: none"> • $256 - s.mask = b.sz$ • $0, 0+b.sz, 0+2*b.sz, \dots s.mask$
4	broadcast addr.	next subnet-1; last is 255
5	valid hosts	from subnet+1 to b.cast-1

Example

Subnetting 192.168.10.0 using /27.
 S.mask 255.255.255.224 (1... 1... 1... 11100 000)
 N.of masked bits: x=3
 N.of unmasked bits y=5
 N.of subnets $2^3=8$

N.of hosts in subnet $2^5-2=30-2=30$
 Block size $256-224=32$

net	b.cast	1st host	lst host
0	31	1	30
32	63	33	62
64	95	65	94
96	127	97	126
128	159	129	158
160	191	161	190
192	223	193	222
224	255	225	254

Appendix: variable length subnetting

Given the base network, for each subnet do the following:

- how many hosts in the subnet? we determine the block size as hosts+2 and searching the minimum block size that contains it;
- from the block size, we get prefix and s.mask. Then we list previous values as follow, allocating subnets from the largest to the smallest block, one after the other, without overlapping

net	hosts	block	prefix	mask	base	subnet
A						
B						
...						

Example. 3 subnets: A, 25 hosts, B 5 hosts, C 2 hosts.
 Base lan: 192.168.10.0

net	hosts	block	prefix	mask		subnet
A	25	$26+2=27$ → 32	/27	224	0	192.168.10.0/ 27
B	5	$5+2=7$ → 8	/29	248	0+32	192.168.10.3 2/29
C	2	$2+2=4$	/30	252	32+8	192.168.10.4 0/30
					40+2	

Document

Author: luciano de falco alfano. Ver: 0.3 of 23rd jun 2019.
 Licenze: [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/). Repository: [cios_cmds on GitHub](https://github.com/luciano-de-falco-alfano/cios_cmds).