CCNA Router command cheat-sheet

Useful router commands for CCNA2 v6 (Routing and Switching Essentials) Cisco Networking Academy course.

Table of contents

Before we start: Configuration modes

Three basic configuration modes we MUST be familiar with already (you will see them below, a lot).

Mode (prompt)	Mode	Mode change (current -> next)
`R1>`	user EXEC mode AKA *view-only mode*	type `enable` to pass to next mode
`R1#`	Privileged EXEC mode	type `configure terminal` to pass to next mode
`R1(config)#`	Global configuration mode	N/A

Common abbreviations to the commands above (separated by commas):

```
en, ena
conf t, config term
```

On a Cisco router, from **global configuration mode**, you can also access the two following configuration modes:

Mode (prompt)	Mode	Description
`R1(config-if)#`	Interface configuration mode	used to configure an individual interface
`R1(config-router)#`	Router configuration mode	used when configuring an IPv4 routing protocol

> 💡 When configuring an IPv6 routing protocol, such as OSPFv3, the router configuration mode prompt will be: `R1(config-rtr)#`

Important `show` commands:

Note that these commands are executed on privileged EXEC mode (`R1#` prompt).
You can execute them from global configuration mode (`R1(config)#` prompt) by adding the `do` keyword before the command.
example:

```
R1(config)#do show ip interface brief
```

Command	Description
`R1#show running-config`	display config saved in RAM (and the currently running configuration in the device)
`R1#show startup-config`	display config saved in NVRAM
`R1#show history`	

Filtering information from `show` commands:

Some commands, such as `show running-config`, generate multiple lines of output.

To filter output, you can use the *pipe* (`|`) character along with a **filtering parameter** and a **filtering expression**.

Filtering parameters	Effect
`section [filtering-expression]`	shows the section of the *filtering expression*
`include [filtering-expression]`	includes all lines of output that match the *filtering expression* **ONLY**
`exclude [filtering-expression]`	excludes all lines of output that match the *filtering expression*
`begin [filtering-expression]`	shows all the lines of output **beginning from** the line that matches the *filtering expression*

Usage:

Here's an example of the usage of filtering with a `show` command:

```
R1#show running-config | include line con
```

💡 ProTip: By default, the screen of output consists of 24 lines. Should you want to change the number of output lines displayed on the terminal screen, you can use the command:

```
R1# terminal length [number-of-lines]
```

⚠️ Unfortunately, this command is NOT supported in Cisco Packet Tracer (tested on version 7.2.2).

Static routing

With an administrative distance (AD) of `1` . These are, just after the **directly connected routes**, the most trustworthy routes a router can have on its routing table.

Static routes are all configured in **global configuration mode**, with the `ip route` command, i.e.,

```
R1(config)#ip route [network-address] [subnet-mask] ....
```

There are essentially four types of static routes, with particular purposes.

Standard Static Route

```
R1(config)#ip route [network-address] [subnet-mask] ([next-hop-ip] [exit-int])
```

Default Static Route

```
R1(config)#ip route 0.0.0.0 0.0.0.0 ([next-hop-ip] [exit-int])
```

Floating Static Route

```
R1(config)#ip route [network-address] [subnet-mask] ([next-hop-ip] [exit-int])
[AD]
```

Summary Static Route

```
R1(config)#ip route [network-address] [subnet-mask*] ([next-hop-ip] [exit-int])
```
* Using the appropriate subnet mask that summarizes the otherwise separate routes into a **single**, static route.

Dynamic routing: OSPF

OSPF configuration

Command	Description
`R1(config)#router ospf [PID]`	create OSPF process with Process ID [PID] (1 - 65535)
`R1(config-router)#router-id [a.b.c.d]`	manually assign the router an ID, in an IPv4 address format.
`R1(config-router)#network [network-address] [wildcard-mask] area [area id]`	for all directly connected networks, announce each network following this nomenclature. Area ID can go from 0 to 4294967295

💡 The `area id` can also be expressed in IP address format, hence the range of available `area id` S.

💡 ProTip: How can I easily visualize all the directly connected networks a router has?
Issue the `R1(config)#do show ip route con` command.
It will display the routing table ONLY with the directly connected networks (routes).

> 🏆 **Best practice:** OSPF routers **within an area** use **and need unique IDs** to identify themselves. It is highly convenient to manually set a desired router ID with the `router-id` command.

OSPF verification

The following are useful commands to verify and troubleshoot your OSPF configurations.

Command	Description
`R1#show ip protocols`	your go-to command to quickly verify key OSPF configuration information on a router, including PID, router ID, advertised networks, neighbors and administrative distance.
`R1#show ip route`	adfa. Additionally, issue `shw ip route ospf` to see only the OSPF-learned networks (entries) on the routing table.
`R1#show ip ospf neighbor`	list of this router's OSPF *neighbor* routers

Command	Description
`R1#show ip ospf`	useful to identify the PID, router ID, area information, and when the SPF algorithm was run for the last time
`R1#show ip ospf interface brief`	summary of router's OSPF-enabled interfaces. ⚠️ This command does NOT seem to work on Packet Tracer 7.3
`R1#show ip ospf interface`	in-detail list of every OSPF-enabled interface
`R1#show ip ospf interface [int-id]`	in-detail information for a particular OSPF-enabled interface

💡 Recall: OSPF's default AD is `110`
You can find common AD values [here](#).

Configuring Router-on-a-stick inter-VLAN routing

To enable router-on-a-stick inter-VLAN routing, we must configure multiple **subinterfaces** (as many as the number of VLANs over which we want to route traffic).
Through this process, only one physical interface, and thus, one trunk link, are required.

In global configuration mode, i.e., `R1(config)#` prompt:

Command	Description
`R1(config)#interface g0/0.[vlan-id]`	create the `.[vlan-id]` **subinterface** on interface Gigabit Ethernet 0/0
`R1(config-subif)#encapsulation dot1q [vlan-id]`	configure subinterface to operate on a specified VLAN
`R1(config-subif)#encapsulation dot1q [vlan-id] native`	must be configured on the subinterfaace belonging to the **native** VLAN
`R1(config-subif)#ip address [ip-address] [subnet-mask]`	❗ don't forget to assign the subinterface an IP address!
`R1(config-subif)#interface g0/0`	access the Gigabit0/0 interface (i.e., the actual physical interface) to enable it

Command	Description
`R1(config-if)#no shutdown`	enable the physical interface. This enables **all** configured subinterfaces **on that interface**

🚷 Access Control Lists (ACL)

💡 ProTip: ***Standard or extended?*** - Standard ACLs are generally -if not always- placed closest to the destination. Extended ACLs are placed closest to the source. Think you might have trouble remembering that? Try this:

> "*Standard closest to Destination*", i.e., **S - D**
> "*Extended closest to Source*", i.e., **E - S**
> Notice (and remember) there is ***only one S*** on each the previous letter pairs.

💡 Recall: ***How many ACLs can I have on the router?*** - One per interface, per protocol (IPv4, IPv6), per direction (inbound, outbound)

Procedure for configuring Standard Numbered ACLs

Command	Description
`R1(config)#access-list [number]` `[permit/deny] [address] [wildcard mask]`	create entry in standard IPv4 ACL with specified number
`R1(config)#interface [int-id]`	select the interface to which the ACL will be applied
`R1(config-if)#ip access-group [number]` `[in/out]`	activate the ACL on the selected interface
`R1(config)#no access-list [number]`	⚠️ This **ERASES** the ACL, i.e., **EVERY ACE** (Access Control entry) **that makes it up**

> 🏆 ⚠️ **Best practice / IMPORTANT:** It is HIGHLY recommended to always have EACH of your ACLs **backed-up.**

Procedure for configuring Standard Named ACLs

Command	Description
`R1(config)#ip access-list [standard] [name]`	creates a standard named ACL. Will take you to standard (`std`) named ACL (`nacl`) configuration mode after entering it
`R1(config-std-nacl)#`	

DHCPv4

Configure a Cisco router as a DHCPv4 server

Command	Description
`R1(config)#ip dhcp excludded-address [start-address (end-address)]`	exclude a single or a range of IPv4 addresses, i.e., these addresses will not be available for DHCP assignment to hosts
`R1(config)#ip dhcp pool [pool name]`	create and name the pool of available addresses to assign. Takes you to **DHCP config mode**.
`R1(dhcp-config)#network [network-id] [mask]`	specify the network address with either the subnet mask OR the prefix length
`R1(dhcp-config)#default-router [address]`	specify the IPv4 address of the default gateway (router) the hosts will use

That's all a simple DHCP server configuration is comprised of. Some additional configuration can be made with help of the follwing commands:

Command	Description
`R1(dhcp-config)#lease [days] ([hours] [minutes])`	specify the time of the lease of every address
`R1(dhcp-config)#dns-server [dns-address-1] ([dns-address-2 ... dns-address-8])`	specify up to 8 DNS server addresses
`R1(dhcp-config)#domain-name [name]`	specify the domain name

The ip helper address:

If your DHCP server is on a **separate network segment**, you will have to (and can) relay external DHCPv4 requests.
You will have to enter the following command on every interface of the router that needs to reach the DHCP server's segment.

Command	Description
`R1(config-if)#ip helper address [DHCP-server-address]`	indicates the router to relay (instead of discarding) DHCP messages to the DHCP server's address.

DHCPv6

Stateless DHCPv6

NAT

Configuring Static NAT

Command	Description
`R1(config)#ip nat inside source static [local-add] [global-add]`	configure static nat, specifying the local address that will be mapped to a global address

Inside and outside interfaces

💡 Recall: After this, **always remember** to specify the **inside** and **outside** interfaces.

> Though it might be easy for us to identify them with our topology diagram, it's not that easy for the router.

Command	Description
`R1(config)#interface [int-id]`	enter configuration mode for the *inside* interface
`R1(config-if)#ip nat inside`	
`R1(config)#interface [int-id]`	enter configuration mode for the *outside* interface

Command	Description
`R1(config-if)#ip nat outside`	

Configuring Dynamic NAT

Command	Description
`R1(config)#ip nat pool [pool-name] [start-address] [end-address] {netmask [mask]	prefix-length [prefix-length]}`	create the pool of routable addresses that hosts will use to exit the private network
`access-list [number] {permit	deny} [source] [wildcard mask]`	create an access list to blah blah
`ip nat inside source list [acl-number] pool [pool name]`	we now bind the pool to the ACL

💡 Lastly, **DO NOT forget to** [specify the inside and outside interfaces](#)

HSRP

Command	Description
`R1(config-if)#standby version 2`	(optional) specify HSRP version 2 will be used. Otherwise, your HSRP instance will be running in version 1 by default.
`R1(config-if)#standby [group] ip [virtual-ip]`	specify the router group and the virtual gateway's IP address
`R1(config-if)#standby [group] priority [0-255]`	(optional) specify each router a priority for **active router** election
`R1(config-if)#standby [group] preempt`	(optional) enables **preemption**, i.e., a router with higher priority will take the active role if current active router's priority is lower
`R1(config-if)#standby [group] timers [hello-timer] [hold-timer]`	specify hello and hold (dead) timers. HSRP automatically ensures that the hold timer value is greater than the hello timer value to avoid incorrect behavior.

⚠️ **ALWAYS use the SAME HSRP version on your groups/topologies**

For Cisco routers, having instances of the 2 different HSRP versions running is like having two different HSRP groups. Hence, you will come across a duplicate address error (assuming you configured HSRPv1 and HSRPv2 using the same virtual router address).

💡 **HSRPv1 vs HSRPv2: What's different?**

- v2 supports IPv6, v1 does NOT.
- v2 supports 4095 groups, v1 supports 255.
- v2 sends multicast HSRP traffic to `224.0.0.102`, v1 multicasts to� `224.0.0.2`
- the **virtual MAC addresses are also different**
  - v1: `0000.0C07.ACxx`
  - v2: `0000.0C9F.Fxxx`
  - 💡 the `x` characters at the end are replaced by the **group number** in HEX.

Appendix: Common administrative distance (AD) values:

Default values:

code	type	AD value
C	Directly connected network	0
S	Static route	1
D	Internal EIGRP	90
O	OSPF	110
R	RIP	120

[Back to table of contents](#)
[Back to OSPF verification](#)
[Back to EIGRP verification](#) [Back to RIP](#)

Appendix: IPv4 address classes

There are five class groups in which IPv4 addresses are segmented: A, B, C, D, and E.
💡 Classes D and E are reserved groups.

- Class D addresses are **multicast groups**

- Class E addresses are reserved for research purposes

Class	Range (first octet)
A	1 - 127
B	128 - 191
C	192 - 223
D	224 - 239
E	240 - 255

[Back to table of contents](#)

Appendix: Private IPv4 address ranges

The following private IPv4 address spaces are defined in RFC 1918 ("Address Allocation for Private Internets").
These addresses are used within private networks and are thus NOT routable to the **public Internet**.
To forward traffic from a private to a public network, you'd use Network Address Translation (NAT), defined in RFC 3022. NAT configuration on Cisco routers is covered [here](#).

Class	Range (CIDR notation)	Range (start - end)
A	10.0.0.0/8	10.0.0.0 - 10.0.0.255
B	172.16.0.0/12	172.16.0.0 - 172.31.255.255
C	192.168.0.0/16	192.168.0.0 - 192.168.255.255

[Back to table of contents](#)

Legacy section (CCNA version 6, 200-125 exam)

⚠️ **Attention:** the following topics are no longer on the **current** version (7) of the 200-301 CCNA exam.

Dynamic routing: RIP

RIPv1 configuration

Command	Description
`R1(config)#router rip`	enter router configuration mode, with RIP as the routing protocol
`R1(config-router)#network [network-id]`	specify the network segment address. Note that this is done without the **subnet mask**.
`R1(config-router)#passive-interface [int-id]`	(optional) - prevent an interface from sending RIP updates

💡 For safety reasons, interfaces that do NOT need to send RIP updates (e.g., interfaces facing network segments with hosts instead of other routers) should be configured as passive. It's also a good practice as RIP updates on these segments would be nothing but wasted bandwidth.

💡 As you see, NO subnet mask information is configured. When forwarding updates, the router uses either the mask configured on the local interface or the default mask based on the address class (A, B, or C). 💡 You can find out more about address classes here
⚠️ Because of this, your networks on a RIPv1 scheme **MUST BE CONTIGUOUS**, meaning that VLSM or supernetting is not supported by version 1 of RIP

💡 Recall: RIP's default AD is `120`
You can find common AD values here.

Additional RIPv1 configuration

Command	Description
`R1(config-router)#default-information originate`	**if a default static route is configured on the router** propagate the default route to other routers receiving RIP updates

RIPv2 configuration

Command	Description
`R1(config)#router rip`	enter router configuration mode, with RIP as the routing protocol
`R1(config-router)#version 2`	change RIP process version from 1 to 2
`R1(config-router)#no auto-summary`	disable network auto summarization

💡 Recall: by default (without previously configuring version 2), RIP can **receive both** RIPv1 and RIPv2 routing updates.
💡 You can see this with the `R1#show ip protocols` command.

Dynamic routing: EIGRP

EIGRP Configuration

Command	Description
`R1(config)#router eigrp [AS-number]`	[AS-number] value range: 1-65535.
`R1(config-router)#eigrp router-id [a.b.c.d]`	(optional) manually configure a **router ID**, in an IPv4 address format
`R1(config-router)#network [network-address]`	add the classful network address for each directly connected network
`R1(config-router)#network [network-address] ([wildcard-mask])`	add the network address with the wildcard mask, recommended when using classless addresing

💡 If no EIGRP router ID is configured, the router will use the highest IPv4 address of any active loopback interface. If the router has NO active loopback interfaces, the router ID will be the highest IPv4 address of any active physical interface.

💡 ProTip: How can I easily visualize all the directly connected networks a router has?
Issue the `R1(config)#do show ip route con` command.
It will display the routing table ONLY with the directly connected networks (routes).

💡 Recall: by default, EIGRP does NOT automatically summarize networks.

EIGRP Verification

Command	Description
`R1#show ip protocols`	verifies the current configured values for EIGRP (and any additional enabled routing protocol on the device)
`R1#show ip eigrp neighbors`	displays the nighbor table. Use it to verify the router recognizes its neighbors.
`R1#show ip route eigrp`	display **only EIGRP** entries in the routing table
`R1#show ip eigrp interface`	verifies which interface are enabled for EIGRP, number of peers, and transmit queues

💡 Recall: (internal) EIGRP's default AD is `90`
You can find common AD values here.

- *The AD for an **EIGRP summary route** is 5.*
- *The AD for an **EIGRP external route** is 170.*

EIGRP Fine tuning

Command	Description
`R1(config-router)#variance [variance]`	change variance to perform **unequal cost load balancing**. [variance] value range: 1-128.
`R1(config-router)#auto-summary`	enable **automatic summarization**
`R1(config-router)#redistribute static`	propagate a **default static route**. Works on both IPv4 and IPv6. 💡 remember to check routing tables to ensure correct/desired configuration
`R1(config-if)#bandwidth [BW in kilobits]`	modify the EIGRP metric. ⚠️ this is done in **interface configuration mode**
`R1(config-if)#ip hello-interval eigrp [AS-number] [time in seconds]`	modify hello interval on IPv4. ⚠️ this is done in **interface configuration mode**
`R1(config-if)#ipv6 hello-interval eigrp [AS-number] [time`	modify hello interval on IPv6. ⚠️ this is done in **interface configuration mode**

Command	Description
`in seconds]`	
`R1(config-if)#ip hold-time eigrp [AS-number] [time in seconds]`	modify hold timer length on IPv4 ⚠️ this is done in **interface configuration mode**
`R1(config-if)#ipv6 hold-time eigrp [AS-number] [time in seconds]`	modify hold timer length on IPv6 ⚠️ this is done in **interface configuration mode**

💡 Recall: **automatic summarization** is **disabled** by default

⚠️ When modifying EIGRP hello intervals and hold timers, ALWAYS MAKE SURE your Hello intervals are LESS THAN your hold timers. Otherwise, you could have a *flapping link* (due to timer misconfiguration, constantly goes up, down, up, ...)

Dynamic routing: EIGRP for IPv6

Configuration

Command	Description
`R1(config)#ipv6 unicast-routing`	this **global configuration command** enables the router to forward IPv6 packets
`R1(config)#ipv6 router eigrp [AS-number]`	enter router configuration mode for EIGRP for IPv6, specifying the AS number
`R1(config-rtr)#eigrp router-id [a.b.c.d]`	💡 The EIGRP for IPv6 process will start running **after** you enter the router ID
`R1(config-rtr)#no shutdown`	💡 This is a best practice, since EIGRP for IPv6 has a **shutdown feature**
`R1(config-rtr)#interface [int-id]`	now, go to **each interface** where you want to enable EIGRP for IPv6
`R1(config-if)#ipv6 eigrp [AS-number]`	use this command with the AS-number you used when configuring EIGRP for IPv6 on **each desired interface**

💡 Note that the configuration mode prompt when configuring EIGRP for IPv6 is different. While EIGRP for IPv4 is configured under the configuration mode with prompt `R1(config-router)#`, EIGRP for IPv6 is configured under the mode/prompt `R1(config-rtr)#`

Verification

The same process for verification or troubleshooting for IPv4 can be used on IPv6 implementations. Replace `ip` for `ipv6` in your commands

Command	Description
`R1#show ipv6 protocols`	verifies the current configured values for EIGRP (and any additional enabled routing protocol on the device)
`R1#show ipv6 eigrp neighbors`	displays the nighbor table. Use it to verify the router recognizes its neighbors.
`R1#show ipv6 route eigrp`	display **only EIGRP** entries in the routing table
`R1#show ipv6 eigrp interface`	verifies which interface are enabled for EIGRP, number of peers, and transmit queues