

## - The Cisco IOS -

### Cisco IOS

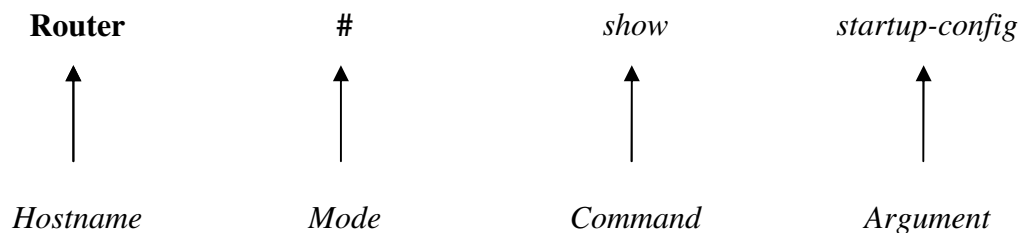
The **Cisco IOS (Internetwork Operating System)** is a command-line interface used by nearly all current Cisco routers and Catalyst switches. The IOS provides the mechanism to configure all Layer 2 and Layer 3 functions on Cisco devices.

The IOS is structured into several **modes**, which contain sets of commands specific to the function of that mode. Access to a specific mode (and specific commands) is governed by **privilege levels**. (Both modes and privilege levels are covered in great detail in this guide).

The following is a representation of the IOS command-line interface, with an example command:

```
Router# show startup-config
```

All commands throughout all guides on this site will be represented like the above. The following is an explanation of each component of the above command:



Hitting the “enter” key after a command will usually yield output specific to your command:

```
Router# show startup-config
!
version 12.2
service timestamps log uptime
service password-encryption
!
hostname Router
!

<snip>
```

(Note: The above output was truncated to save space.)

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Version Numbers

IOS version numbers are formatted as follows:

$$x.y(z)t$$

- The "x" designates a **major revision number**.
- The "y" designates a **minor revision number**.
- The "z" designates an **individual release number**
- The "t" designates a **train identifier**.

Thus, the third release of IOS version 12.4 would be identified as 12.4(3). The major and minor revision numbers combined is often called the **Maintenance Release** number (for example, "12.4").

**Trains** identify IOS releases to specific markets, and are represented by a single letter:

- The "T" or **Technology** train is continuously updated with new features and security fixes.
- The "E" or **Enterprise** train contains features and commands for enterprise-level equipment.
- The "S" or **Service Provider** train contains features and a command-set for specific ISP equipment

The absence of a train identifier denotes a **Mainline** release. Security updates are released for the mainline train, but new functionality is never added to the feature set.

The latest version of the IOS (as of this writing) is 12.4(11)T. To view the IOS version of your Cisco device:

```
Router# show version
```

The Cisco IOS is stored in **Flash** on Cisco routers and Catalyst switches, in a **.BIN** file format. It can be upgraded using one of several methods:

- Replacing the existing Flash stick
- Via a TFTP server
- Via Xmodem
- Via a PCMCIA slot (not supported by all Cisco devices)

(Reference: [http://en.wikipedia.org/wiki/Cisco\\_IOS](http://en.wikipedia.org/wiki/Cisco_IOS))

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

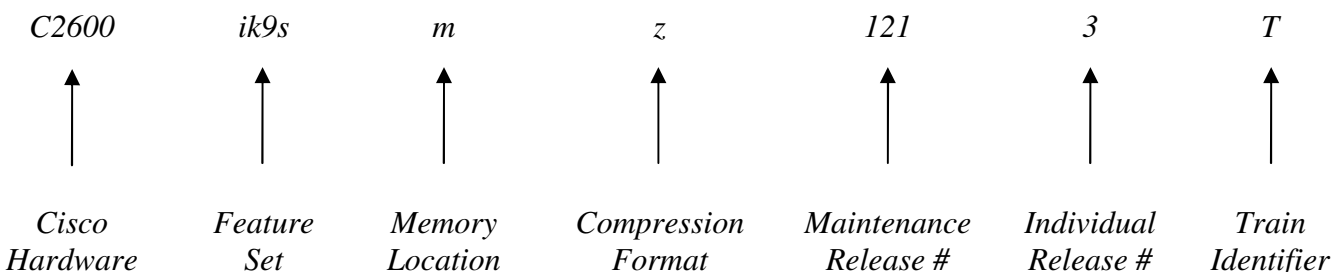
This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Version Numbers (continued)

The IOS .bin file stored in flash follows a specific naming convention. Observe the following IOS image:

c2600-ik9s-mz.121-3.T.bin

The following is an explanation of each component of the above file name:



(Reference: [http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products\\_white\\_paper09186a008018305e.shtml](http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml))

The IOS supports a wide variety of feature sets. The following is a list of common feature sets (and is by no means comprehensive):

- **is**
- **ipbase**
- **ipvoice**
- **advsecurityk9**
- **advipservicesk9**
- **ik9s**
- **jk9s**
- **io3**
- **bin**

(Reference: [http://www.cisco.com/en/US/products/hw/routers/ps259/prod\\_bulletin09186a0080161082.html](http://www.cisco.com/en/US/products/hw/routers/ps259/prod_bulletin09186a0080161082.html))

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Modes on Cisco Devices

As stated earlier in this guide, the Cisco IOS is comprised of several modes, each of which contains a set of commands specific to the function of that mode.

By default, the first mode you enter when logging into a Cisco device is **User EXEC mode**. User mode appends a “>” after the device hostname:

```
Router>
```

No configuration can be changed or viewed from User mode. Only basic status information can be viewed from this mode.

**Privileged EXEC mode** allows all configuration files, settings, and status information to be viewed. Privileged mode appends a “#” after the device hostname:

```
Router#
```

To enter Privileged mode, type *enable* from User mode:

```
Router> enable
Router#
```

To return back to User mode from Privileged mode, type *disable*:

```
Router# disable
Router>
```

Very little configuration can be *changed* directly from Privileged mode. Instead, to actually configure the Cisco device, one must enter **Global Configuration mode**:

```
Router(config)#
```

To enter Global Configuration mode, type *configure terminal* from Privileged Mode:

```
Router# configure terminal
Router(config)#
```

To return back to Privileged mode, type *exit*:

```
Router(config)# exit
Router#
```

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Modes on Cisco Devices (continued)

As its name implies, Global Configuration mode allows parameters that *globally* affect the device to be changed. Additionally, Global Configuration mode is sectioned into several sub-modes dedicated for specific functions. Among the most common sub-modes are the following:

- **Interface Configuration mode** - **Router(config-if)#**
- **Line Configuration mode** - **Router(config-line)#**
- **Router Configuration mode** - **Router(config-router)#**

Recall the difference between *interfaces* and *lines*. **Interfaces** connect routers and switches to each other. In other words, traffic is actually routed or switched across interfaces. Examples of interfaces include Serial, ATM, Ethernet, Fast Ethernet, and Token Ring.

To configure an interface, one must specify both the *type* of interface, and the interface *number* (which always begins at “0”). Thus, to configure the first Ethernet interface on a router:

```
Router(config)# interface ethernet 0
Router(config-if)#
```

**Lines** identify ports that allow us to connect into, and then configure, Cisco devices. Examples would include console ports, auxiliary ports, and VTY (or telnet) ports.

Just like interfaces, to configure a line, one must specify both the *type* of line, and the line *number* (again, always begins at “0”). Thus, to configure the first console line on a router:

```
Router(config)# line console 0
Router(config-line)#
```

Multiple telnet lines can be configured simultaneously. To configure the first five telnet (or VTY) lines on a router:

```
Router(config)# line vty 0 4
Router(config-line)#
```

Remember that the numbering for both interfaces and lines begins with “0.”

**Router Configuration** mode is used to configure dynamic routing protocols, such as RIP. This mode is covered in great detail in other guides.

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Command Shortcuts

**Shortcuts** are allowed on the IOS command-line, as long as the truncated command is not ambiguous. For example, observe the following commands:

```
Router# clear
Router# clock
Router# configure
Router# connect
Router# copy
Router# debug
```

We could use *de* as a shortcut for the *debug* command, as no other command here begins with *de*. We could not, however, use *co* as a shortcut, as three commands begin with those letters. The following error would be displayed:

```
Router# co
% Ambiguous command: "co"
```

If you type a command incorrectly, the IOS will point out your error:

```
Router# clcok
      ^
% Invalid input detected at "" marker
```

## Keyboard Shortcuts

Several hotkeys exist to simplify using the IOS interface:

<u>Keyboard Shortcut</u>	<u>Result</u>
CTRL-B (or Left-Arrow)	<i>Moves cursor back one character</i>
CTRL-F (or Right-Arrow)	<i>Moves cursor forward one character</i>
CTRL-A	<i>Moves cursor to beginning of a line</i>
CTRL-E	<i>Moves cursor to end of a line</i>
ESC-B	<i>Moves cursor back one word</i>
ESC-F	<i>Moves cursor forward one word</i>
CTRL-P (or Up-Arrow)	<i>Returns previous command(s) from history buffer</i>
CTRL-N (or Down-Arrow)	<i>Returns next command from history buffer</i>
CTRL-Z	<i>Exits out of the current mode</i>
TAB	<i>Finishes an incomplete command (assuming it is not ambiguous)</i>

(Reference: [http://www.cisco.com/en/US/products/hw/switches/ps708/products\\_configuration\\_guide\\_chapter09186a008007e6d5.html#wp1028871](http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_guide_chapter09186a008007e6d5.html#wp1028871))

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## **Terminal History Buffer**

As implied in the previous section, the Cisco IOS keeps a **history** of previously entered commands. By default, this history buffer stores the previous **10** commands entered. To view the terminal history buffer:

```
RouterA# show history

enable
config t
hostname RouterA
exit
show history
```

The **Up-Arrow** key (or **CTRL-P**) allows you to scroll through previously entered commands. To scroll back down the list, use the **Down-Arrow** key (or **CTRL-N**).

To adjust the number of commands the history buffer stores (range 0-256):

```
RouterA# terminal history size 30
```

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Context-Sensitive Help

The **question mark (?)** is one of the most powerful tools in the Cisco IOS, as it provides **context-sensitive help** for each IOS mode.

Typing **?** at the command prompt displays a list of all commands available at that mode, with explanations:

**Router# ?**

access-enable	Create a temporary Access-List entry
access-profile	Apply user-profile to interface
access-template	Create a temporary Access-List entry
alps	ALPS exec commands
archive	manage archive files
audio-prompt	load ivr prompt
bfe	For manual emergency modes setting
call	Load IVR call application
cd	Change current directory
clear	Reset functions
clock	Manage the system clock
configure	Enter configuration mode
connect	Open a terminal connection
copy	Copy from one file to another
debug	Debugging functions (see also 'undebug')

<snip>

Typing in part of a command with a **?** displays a list of all commands that begin with those characters:

**Router# co?**

configure connect copy

Typing in a full command followed by a **?** displays the available options and arguments for that command:

**Router# clock ?**

set Set the time and date

Notice the space between the command *clock* and the **?**.

\*\*\*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.



## The “Show” Command

The *show* command provides the ability to view a wide variety of configuration and status information on your router. The command is executed from Privileged mode, and the syntax is simple:

```
Router# show [argument]
```

There are literally dozens of arguments for the *show* command, and each provides information on a specific aspect of the router. Numerous *show* commands will be described throughout this and most other guides.

One common *show* command displays the IOS version, configuration-register settings, router uptime, and basic router hardware information:

```
Router# show version
```

```
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-IS-L), Version 12.3(1a), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Fri 09-Jan-03 11:23 by xxxxx
Image text-base: 0x0307F6E8, data-base: 0x00001000

ROM: System Bootstrap, Version 11.0(10c)XB2, PLATFORM SPECIFIC RELEASE SOFTWARE
(fc1)
BOOTLDR: 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c)XB2, PLATFORM
SPECIFIC RELEASE SOFTWARE (fc1)

Router uptime is 2 minutes
System returned to ROM by reload
System image file is "flash:c2500-is-1.123-1a.bin"

cisco 2500 (68030) processor (revision L) with 14336K/2048K bytes of memory.
Processor board ID 13587050, with hardware revision 00000000
Bridging software.
X.25 software, Version 3.0.0.
2 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

Configuration register is 0x2102
```

(Example *show version* output from: [http://www.cisco.com/en/US/products/hw/routers/ps233/products\\_tech\\_note09186a008009464c.shtml](http://www.cisco.com/en/US/products/hw/routers/ps233/products_tech_note09186a008009464c.shtml))

The following command provides output similar to *show version*:

```
Router# show hardware
```

Other common *show* commands will be described shortly.

\*\*\*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)),  
unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Enable Passwords

The *enable* password protects a router's Privileged mode. This password can be set or changed from Global Configuration mode:

```
Router(config)# enable password MYPASSWORD
Router(config)# enable secret MYPASSWORD2
```

The *enable password* command sets an unencrypted password intended for legacy systems that do not support encryption. It is no longer widely used.

The *enable secret* command sets an MD5-hashed password, and thus is far more secure. The *enable password* and *enable secret* passwords **cannot be identical**. The router will not accept identical passwords for these two commands.

## Line Passwords and Configuration

Passwords can additionally be configured on router **lines**, such as telnet (vty), console, and auxiliary ports. To change the password for a console port and all telnet ports:

```
Router(config)# line console 0          Router(config)# line vty 0 4
Router(config-line)# login             Router(config-line)# login
Router(config-line)# password cisco1234 Router(config-line)# password cisco1234

Router(config-line)# exec-timeout 0 0  Router(config-line)# exec-timeout 0 0
Router(config-line)# logging synchronous Router(config-line)# logging synchronous
```

The *exec-timeout 0 0* command is optional, and disables the automatic timeout of your connection. The two zeroes represent the timeout value in minutes and seconds, respectively. Thus, to set a timeout for 2 minutes and 30 seconds:

```
Router(config-line)# exec-timeout 2 30
```

The *logging synchronous* command is also optional, and prevents system messages from interrupting your command prompt.

By default, line passwords are stored in clear-text in configuration files. To ensure these passwords are encrypted in all configuration files:

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

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Router Interfaces

Recall that, to configure an interface, one must specify both the *type* of interface, and the interface *number* (which always begins at “0”). Thus, to configure the first Ethernet interface on a router:

```
Router(config)# interface ethernet 0
Router(config-if)#
```

Certain router families (such as the 3600 series) are modular, and have multiple “slots” for interfaces. All commands must reflect both the **module number** and the interface number, formatted as: *module/interface*

Thus, to configure the third Fast Ethernet interface off of the first module:

```
Router(config)# interface fastethernet 0/2
Router(config-if)#
```

By default, all router interfaces are **administratively shutdown**. To take an interface out of an administratively shutdown state:

```
Router(config)# interface fa 0/0
Router(config-if)# no shutdown
```

Notice the use of *fa* as a shortcut for *fastethernet* in the above example. To manually force an interface into a shutdown state:

```
Router(config-if)# shutdown
```

To assign an IP address to an interface:

```
Router(config-if)# ip address 192.168.1.1 255.255.255.0
```

An additional *secondary* IP Address can be assigned to an interface:

```
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# ip address 192.168.1.2 255.255.255.0 secondary
```

Serial interfaces require special consideration. The **DCE (Data Communication Equipment)** side of a serial connection must set the speed, or *clock rate*, for the **DTE (Data Terminal Equipment)** side. *Clock rate* is measured in BPS (bits-per-second).

To set the *clock rate*, if you are the DCE side of a serial connection:

```
Router(config)# interface serial 0
Router(config-if)# clock rate 64000
```

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Status of Router Interfaces

To view the current status and configuration of all interfaces:

**Router#** *show interfaces*

```

Ethernet 0 is up, line protocol is up
  Hardware is Ethernet, address is 5520.abcd.1111
  Internet address is 192.168.1.1, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 2000 bits/sec, 4 packets/sec
    53352 packets input, 351251 bytes, 0 no buffer
    Received 4125 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    12142 packets output, 16039 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets, 0 restarts

Ethernet 1 is up, line protocol is up
  Hardware is Ethernet, address is 5520.abcd.1112

<snip>

```

The *show interfaces* command displays a plethora of information, including:

- Current interface status (*ethernet 0 is up, line protocol is up*)
- MAC address (*5520.abcd.1111*)
- IP address (*192.168.1.1*)
- MTU (*1500 bytes*)
- Bandwidth (*10 Mbps*)
- Output and input queue status
- Traffic statistics (*packets input, packets output, collisions, etc.*)

To view the current status of a *specific* interface:

**Router#** *show interfaces ethernet 0*

To view only IP information for all interfaces:

**Router#** *show ip interface brief*

Interface	IP Address	OK?	Method	Status	Protocol
Ethernet0	192.168.1.1	YES	NVRAM	up	up
Ethernet1	192.168.2.1	YES	NVRAM	up	up
Serial0	unassigned	YES	unset	administratively down	down

\*\*\*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

**Status of Router Interfaces (continued)**

Traffic can only be routed across an interface if its status is as follows:

***Serial 0 is up, line protocol is up***

The first part of this status (*Serial0 is up*) refers to the **physical layer** status of the interface. The second part (*line protocol is up*) refers to the **data-link layer** status of the interface. A status of *up/up* indicates that the physical interface is active, and both sending and receiving keepalives.

An interface that is physically down will display the following status:

***Serial 0 is down, line protocol is down***

The mostly likely cause of the above status is a defective (or unplugged) cable or interface.

There are several potential causes of the following status:

***Serial 0 is up, line protocol is down***

Recall that *line protocol* refers to data-link layer functions. Potential causes of the above status could include:

- Absence of keepalives being sent or received
- Clock rate not set on the DCE side of a serial connection
- Different encapsulation types set on either side of the link

An interface that has been administratively shutdown will display the following status:

***Serial 0 is administratively down, line protocol is down***

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Managing Configuration Files

Cisco IOS devices employ two distinct configuration files

- **running-config** – stored in RAM, contains the *active* configuration
- **startup-config** – stored in NVRAM (or flash), contains the *saved* configuration

Any configuration change made to an IOS device is made to the running-config. Because the running-config file is stored in RAM, the contents of this file will be lost during a power-cycle. Thus, we must save the contents of the running-config to the startup-config file. We accomplish this by using the *copy* command from Privileged mode:

```
Router# copy running-config startup-config
```

The copy command follows a very specific logic: *copy [from] [to]*. Thus, if we wanted to copy the contents of the startup-config file to running-config:

```
Router# copy startup-config running-config
```

We can use shortcuts to simplify the above commands:

```
Router# copy run start
```

```
Router# copy start run
```

To view the contents of the running-config and startup-config files:

```
Router# show run
```

```
Router# show start
```

To delete the contents of the startup-config file:

```
Router# erase start
```

If the router is power-cycled after erasing the startup-config file, the router will enter **Initial Configuration Mode** (sometimes called **Setup Mode**). This mode is a series of interactive questions intended for quick reconfiguration of the router.

Initial Configuration Mode can be exited by typing CTRL-C.

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## “Piping” Commands

In newer versions of the Cisco IOS, the output of *show* commands can be filtered to remove irrelevant lines, through the use of the **pipe** “|” character.

The following command will display the contents of the startup-config, **beginning** with the first line containing the text *ethernet*:

```
Router# show startup | begin ethernet
```

The following command will **exclude** all lines containing the text *ethernet*:

```
Router# show startup | exclude ethernet
```

The following command will **include** all lines containing the text *ethernet*:

```
Router# show startup | include ethernet
```

## Miscellaneous Commands

To change the hostname of your router:

```
Router(config)# hostname MyRouter
MyRouter(config)# hostname MyRouter
```

To assign a description to an interface for documentation purposes:

```
Router(config)# interface serial 0
Router(config-if)# description SBC T1 connection to Chicago
```

```
Router# show interfaces
```

```
Serial 0 is up, line protocol is up
Hardware is Serial
Internet address is 70.22.3.1, subnet mask is 255.255.255.0
Description: SBC T1 connection to Chicago
```

To create a *banner* message which users will see when logging into an IOS device:

```
Router(config)# banner motd #
```

```
Logging into this router without authorization is illegal
and will be prosecuted!
#
```

The # symbol is used as a delimiter to indicate the beginning and end of the banner. Any character can be used as a delimiter.

\*\*\*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## IOS Troubleshooting Commands

The *show tech-support* command prints to screen every configuration file, and the output of several important *show* commands. This can be redirected to a file and either viewed or sent to Cisco for troubleshooting purposes:

```
Router# show tech-support
```

The *debug* command is a powerful tool to view real-time information and events occurring on an IOS device. As with the *show* command, there are a multitude of arguments for the *debug* command. An example *debug* command is as follows:

```
Router# debug ip rip events
```

To disable a specific debugging command, simply prepend the word *no* in front of the command:

```
Router# no debug ip rip events
```

To enable all possible debugging options on an IOS device:

```
Router# debug all
```

Using the *debug all* command is not recommended, as it will critically impair router performance.

To disable all possible debugging options on an IOS device:

```
Router# no debug all
```

\* \* \*

All original material copyright © 2007 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.