



5

Basic Switch and Router Configuration

CERTIFICATION OBJECTIVES

- | | | | |
|------|----------------------------|------|----------------------------|
| 5.01 | IOS Introduction | 5.04 | Basic Router Configuration |
| 5.02 | IOS Basics | ✓ | Two-Minute Drill |
| 5.03 | Basic Switch Configuration | Q&A | Self Test |

This chapter presents the basics of configuring the 1900 and 2950 Catalyst switches as well as Cisco routers. As you continue throughout this book, you will build upon these fundamental and important concepts for accessing, configuring, and managing your Cisco devices. I am making the assumption that you have never configured a Cisco device before and therefore will begin with the very basics by explaining the operating system that these devices use, the advantages that Cisco's operating system provides, and how to use some basic operating system commands to configure your Cisco device.

CERTIFICATION OBJECTIVE 5.01

IOS Introduction

One of the main reasons that Cisco is number one in the enterprise networking market place is their *Internetwork Operating System (IOS)*. The IOS provides a similar function to Microsoft Windows XP or Linux: it controls and manages the hardware it is running on. Basically, the IOS provides the interface between you and the hardware, enabling you to execute commands to configure and manage your Cisco device. Originally, the IOS was developed for Cisco routers, but over the last few years, Cisco has been porting the IOS to its other platforms, including the Catalyst switches.

Cisco has spent many years tweaking and tuning the IOS, as well as adding features as new technologies are introduced to the marketplace. Advantages of the IOS include:

- **Features** The IOS includes a wide array of features for protocols and functions that provide connectivity, scalability, reliability, and security solutions for networks of any size.
- **Connectivity** The IOS supports a variety of data link layer technologies for the LAN and WAN environments, including copper and fiber wiring as well as wireless.
- **Scalability** The IOS supports both fixed and modular chassis platforms, enabling you to purchase the appropriate hardware for your needs, yet still allowing you to leverage the same IOS CLI to reduce your management costs.
- **Reliability** To ensure that your critical resources are always reachable, Cisco has developed many products and IOS features to provide network redundancy.
- **Security** With the IOS, you can strictly control access to your network and networking devices in accordance with your internal security policies.

exam

Watch

IOS stands for Internetwork Operating System. Its advantages include features, connectivity, scalability, reliability, and security.

Because of the success of their IOS software, Cisco has grown from a garage-based router company to one of the largest companies in the world in a little over a decade. Most enterprise networks, as well as ISPs, use Cisco products in one form or another. Actually, a large portion of the Internet backbone is composed of Cisco products. With the IOS coupled with a first-class

service and support team, few companies can compare to Cisco when it comes to customer satisfaction.

Device Startup

There are actually many ways of accessing a Cisco device, including the following: console, auxiliary (only certain Cisco routers), telnet, web browser, and an SNMP management station. A console interface provides serial connection access to a router—with console access, you can enter commands in a text-based mode. In order to access your Cisco device from a remote station, however, you first need to create a basic configuration, including IP addressing. Therefore, to perform your initial configurations, you need access to the console port of your Cisco device.

Before you can actually begin configuring your Cisco device, you first have to connect it to your network and set up a terminal connection to its console interface, as described in the Chapter 4. Here are the three steps your Cisco device goes through when booting up:

1. Perform hardware tests.
2. Locate and load the IOS.
3. Locate and execute the device's configuration file.

Once you power on your Cisco device, hardware tests are performed to ensure that it is operating correctly. These tests, *power-on self tests (POST)*, are discussed in Chapter 4. After these tests have completed, the Cisco device finds and locates the IOS and then proceeds to load it. Once the IOS is loaded, the IOS then searches for the device's configuration and executes it. With steps 2 and 3, there are typically fallback measures that the Cisco device goes through if it cannot find an IOS or locate a configuration file.

As you will see in this chapter, and as you work with Cisco devices in a production environment, each Cisco product is unique and may have its own methods for finding and loading its IOS and configuration file. As an example, a Catalyst 1900 switch,

exam

Watch

When an IOS device boots up, it runs P (this doesn't look like it's complete—should there be more here?)

direct from Cisco, comes with a default configuration already on it. This configuration is enough to allow the switch to perform basic switching functions right out of the box. A Cisco router, by contrast, requires some basic configuration in order to route traffic between interfaces.

Accessing the Command-Line Interface (CLI)

How you access the IOS CLI on a Cisco device for the first time depends on the kind of device that you are configuring. In almost every case, you will use the console interface to initially interact with the device; however, gaining access to the CLI from the console port can be different from one device to another. On a Cisco router, for instance, you are taken directly to the IOS CLI when you log in from the console port. If the IOS cannot find a configuration file for the router, the IOS takes you through *Setup* mode, which is a basic configuration script that prompts you for information on how you want to configure your router. However, on a Catalyst 1900 switch, you are first taken to a menu-based interface where you must choose the option of configuring the switch from the CLI. And a 1900 never takes you through *Setup* mode unless you physically execute the command from the CLI to begin the script.

exam

Watch

Be familiar with the various types of access to the IOS CLI: console and auxiliary ports for local access and VTYS (telnet), TFTP, SNMP, and web browsers.

Once you have configured your Cisco device via the console port, you can then use other methods of accessing and changing its configuration, such as telnet—Cisco calls this *virtual terminal (VTY)*—TFTP, SNMP, or a web browser. Cisco has a variety of management

products to configure and manage your Cisco device, such as CiscoWorks 2000; however, the focus of this book is strictly on the IOS CLI.

IOS Differences

What you will see in this and consequent chapters is that even though both Cisco routers and switches run the IOS, the commands used by these products are frequently different! In other words, how you configure a feature on a Cisco router might be, and probably is, different than configuring the same feature on a Catalyst switch. This is even

true among Cisco's Catalyst switches: how you configure a certain feature on a 1900 may be different from configuring the same feature on the 2950!

However, you access the IOS and maneuver around the IOS access levels, as well as using many of the management commands, the same way on *all* IOS products. This can become confusing to a Cisco novice, where one command for a particular feature is the same on all Cisco products, but configuring another feature might be different on a Cisco router than on a Catalyst switch.

Besides command differences between different products, such as routers and switches, there may be command differences within a product line, Cisco routers, for instance. As an example, Cisco sells different flavors of its IOS software for routers, depending on the features that you need. Or because of hardware differences, some commands work on some routers but not on others.

Interacting with the IOS

The CLI is a character- or text-based interface. To interact with the CLI, you only need to type in commands, just as you would do when typing an essay in a text editor or a message in an e-mail program. You can even use functions like cut-and-paste with the IOS CLI: you can copy the complete configuration of a router using a terminal emulator's copy function, paste this into a text editor, make changes to the configuration, select and copy the new configuration, and paste all of these commands back into the CLI.

The CLI supports a command parser. Whenever you press the ENTER key, the IOS parses the command and parameters that you entered and checks for correct syntax and options. When you paste multiple commands into the CLI, the IOS still performs this process for each command that is included in the paste function. If you made a mistake with one command, the CLI parser will display an error message, but continue with the next command in the pasted list.

EXEC Modes

Each Cisco device supports different access modes. For CLI interaction, there are actually three modes:

- *User EXEC* Provides basic access to the IOS, with limited command availability (basically simple monitoring and troubleshooting commands)
- *Privilege EXEC* Provides high-level management access to the IOS, including all commands available at *User EXEC* mode
- *Configuration* Allows configuration changes to be made to the device

6 Chapter 5: Basic Switch and Router Configuration

Both *EXEC* modes can be password-protected, allowing you to limit the people who can access your device to perform management, configuration, and troubleshooting tasks. The next two sections cover the two *EXEC* modes.

User EXEC Mode

Your initial access to the CLI is via *User EXEC* mode, which has only a limited number of commands that you can execute. Depending on the Cisco device's configuration, you might be prompted for a password to access this mode. This mode is typically used for basic troubleshooting of networking problems. You can tell that you are in *User EXEC* mode by examining the prompt on the left-hand side of the screen:

```
Router>
```

If you see a “>” character at the end of the information, then you know that you are in *User EXEC* mode. The information preceding the “>” is the name of the Cisco device. For instance, the default name of all Cisco routers is “Router,” whereas on the 1900 switch, there is no name by default, and you see only the “>” sign at the beginning of the line, like this:

```
>
```

The 2950 switch's *User EXEC* prompt looks like this: `Switch>`. These device names can be changed with the **hostname** command, which is discussed later in this chapter.

Privilege EXEC Mode

Once you have gained access to *User EXEC* mode, you can use the **enable** command to access *Privilege EXEC* mode:

```
Router> enable
Router#
```

Once you enter the **enable** command, if a *Privilege EXEC* password has been configured on the Cisco device, you will be prompted for it. Upon successfully authenticating, you will be in *Privilege EXEC* mode. You can tell that you are in this mode by examining the CLI prompt. In the preceding code example, notice that the “>” changed to a “#.”

When you are in *Privilege EXEC* mode, you have access to all of the *User EXEC* commands as well as many more advanced management and troubleshooting commands. These commands include extended ping and trace abilities, managing configuration files and IOS images, and detailed troubleshooting using **debug** commands. About the only thing that you can't do from this mode is to change the configuration of the Cisco device—this can only be done from *Configuration* mode.

If you wish to return to *User EXEC* mode, from *Privilege EXEC* mode, use the **disable** command:

```
Router# disable
Router>
```

Again, by examining the prompt, you can tell that you are now in *User EXEC* mode.

Logging Out of Your Device

You can log out of your Cisco device from either *User* or *Privilege EXEC* mode by using the **logout** or **exit** command:

```
Router# logout
-or-
Router# exit
```



The Catalyst IOS-based switches do not support the *logout* command, but they do support the *exit* command.



5.01. The CD includes a multimedia demonstration of logging in and out of a Cisco router.

exam

Watch

Know the three different modes in the IOS—User EXEC, Privilege EXEC, and Configuration modes—and what you can do in each mode. Use the *enable* command to go from User EXEC mode

to Privilege EXEC mode. Use the *disable* command to go from Privilege EXEC to User EXEC modes and use the *exit* command to log out of the IOS device from either of these two modes.

IOS Basics

Now that you know how to log in to and out of a Cisco device, the next few sections describe some of the features built into the CLI that will make your configuration and management tasks easier. These features include how to abbreviate commands, how to bring up detailed help on commands and their specific parameters, the output of commands, recalling commands, and editing commands.

Command Abbreviation and Completion

The CLI of the IOS allows you to abbreviate commands and parameters to their most unique characters. This feature is very useful for those of us that are physically challenged at typing. As an example, you could type **en** instead of **enable** when you want to go from *User EXEC* to *Privilege EXEC* mode, like this:

```
Router> en
Router#
```

The Cisco device, internally, completes the command for you. However, the characters that you enter must make the command unique. As an example, you couldn't type just the letter *e*, since there are other commands that begin with the letter *e*, such as **exit**.

Context-Sensitive Help

One of the more powerful features of the IOS is the support of context-sensitive help. Context-sensitive help is supported at all modes within the IOS, including *User EXEC*, *Privilege EXEC*, and *Configuration* modes. There are a variety of ways to use this feature. If you are not sure what command you need to execute, at the prompt, type either **help** or **?**. The Cisco device then displays a list of commands that can be executed at the level in which you are currently located, along with a brief description of each command. Here is an example from a router's CLI at *User EXEC* mode:

```
Router> ?
Exec commands:
  access-enable  Create a temporary Access-List entry
  cd             Change current device
  clear         Reset functions
  connect       Open a terminal connection
  dir           List files on given device
  disable       Turn off privileged commands
```

```

disconnect      Disconnect an existing network connection
enable          Turn on privileged commands
<--output omitted-->
-- More --

```

exam

Watch

Use the `help` command or the `?` to pull up context-sensitive help. Also, you can abbreviate commands to their most unique characters.

If you see “-- More --” at the bottom of the screen, this indicates that there is more help information than can fit on the current screen. On a Cisco device, if you press the SPACEBAR, the IOS pages down to the next screen of help information. On Cisco routers, if you hit the ENTER key, the help scrolls down one line at a time. Any other keystroke breaks out of the help

text. Please note that on the Catalyst switches, the ENTER key is considered a break and terminates the help.

For more detailed help, you can follow a command or parameter with a space and a `?`. This causes the CLI to list the available options or parameters that are included for the command. For instance, you could type **erase** followed by `?` to see all of the parameters available for the **erase** command:

```

Router# erase ?
/all          Erase all files(in NVRAM)
 flash:      Filesystem to be erased
 nvram:      Filesystem to be erased
 pram:       Filesystem to be erased
 slot0:      Filesystem to be erased
 slot1:      Filesystem to be erased
 startup-config Erase contents of configuration memory
Router# erase

```

In this example, you can see at least the first parameter necessary after the **erase** command. Please note that there may be additional parameters, depending on the next parameter that you type.

Or if you’re not sure how to spell a command, you can enter the first few characters and immediately follow these characters with “?”; **e?**, for instance, lists all of the commands that begin with **e** at the current mode:

```

Router# e?
enable erase exit
Router# e

```

In this example, there are three commands that begin with the letter **e** at *Privilege EXEC* mode.



5.02. The CD includes a multimedia demonstration of using context-sensitive help on a Cisco router.

Command Output

Whenever you enter a command, there is chance that you have entered it incorrectly. If this is the case, the IOS tells you that there is a problem with the previously executed command. For instance, this message indicates a CLI input error:

```
% Invalid input detected at '^'.
```

What is important is to examine the line between the command that you typed in and the error message. Somewhere in this line, you'll see an “^” sign. This is used by the IOS to indicate that an error exists in the command line at that spot.

Here is another CLI error message:

```
% Incomplete command.
```

This error indicates that you have not entered all of the necessary parameters for the command: The syntax of the command is correct, but more parameters are necessary. You can use the context-sensitive help feature discussed earlier in this chapter to help you figure out what parameter or parameters you forgot.

You get the following error message if you do not type enough characters to make a command or parameter unique. Here is an example:

```
% Ambiguous command: "show i"
```

In this example, apparently, more than one parameter for the **show** command begins with the letter **i**. Again, you can use context-sensitive help to figure out what parameter to use:

```
Router# show i?  
idb interfaces ip ipv6  
Router# show i
```

If you enter a command that the IOS does not understand, you'll see this error message:

```
% Unknown command or computer name, or unable to find computer address
```

If you see this, use the context-sensitive help in order to figure out the correct command to enter.

exam**Watch**

I cannot begin to stress how important the context-sensitive help is. I constantly use it to enter commands. As of the writing of this book, this feature

is available in the router and switch simulation questions of the exam. Don't be afraid to take advantage of this feature during the simulation questions on the exam.

Entering Commands

Four key features are included in the IOS relate to entering commands:

- Symbolic translation
- Command prompting
- Syntax checking
- Command recall

Whenever you enter a command in the CLI, the command-line parser dissects the command, making sure that it is a valid command with valid parameters. In the case of Cisco routers only, if the CLI parser cannot find the actual command, the IOS assumes that you are trying to telnet to a machine by that name and attempts a DNS resolution of the name to an IP address. This process, called *symbolic translation*, can be annoying at times. But it does make telnetting to a remote machine much easier, since you only have to type the name or IP address of the machine instead of using the **telnet** command, discussed in the next chapter.

You have already been presented with the *command prompting* feature—this is most commonly seen when using the context-sensitive help, like the following:

```
Router# show ?
aaa                               Show AAA values
  aal2                             Show commands for AAL2
  access-expression                List access expression
  access-lists                     List access lists
  accounting                       Accounting data for active sessions
  adjacency                        Adjacent nodes
  alarm-interface                  Display information about a specific Alarm
                                   Interface Card
  aliases                          Display alias commands
  alps                              Alps information
  arp                               ARP table
<--output omitted-->
Router# show
```

In this example, after you use the context-sensitive help with the **show** command, the command **show** is left on the command line after the displayed output. The IOS is assuming that you are entering one of the parameters of this command and thus reenters the command in the CLI. This can be annoying if you, like myself, forget that the router is performing this function and re-enter the command again, like this:

```
Router# show show
```

The CLI always parses your commands and checks their validity by using the *syntax checking* feature. Any nonexistent commands or improperly entered commands cause the IOS to generate an error message with an appropriate error description.



Whenever you enter a command correctly, very rarely will you see any output from the IOS, unless the command you typed in somehow changes the state of the router or one of its components, like an interface coming up or going down. Therefore, you should worry only when you enter a command and the IOS displays a message afterward—then you should assume that there might be a problem.

Of course, when you are configuring a Cisco router or Catalyst switch, you will typically make typing mistakes or enter invalid commands. If you typed in a 20-parameter command and made a mistake with the very last character, it would be sadistic on Cisco's part to have you type the complete command again. The *command recall* feature is an extremely useful one that allows you to recall and edit previously executed commands. The next two sections discuss how to recall and edit previous (or current) commands.

Command-Line History

On any IOS device, use the **show history** command to see your previous commands:

```
Router# show history
enable
  show interface
  show version
  show history
Router#
```

By default, an IOS device stores the last ten commands that you executed. You can recall these commands by pressing either CTRL-P or the UP ARROW key. If you accidentally go past the command that you want to edit or reexecute, use CTRL-N or the DOWN ARROW key.

On IOS routers, you can increase the size of the history buffer from 10 commands up to 256 by using **terminal history size** command:

```
Router# terminal history size #_of_commands
```

Unfortunately, this command is not supported on the 1900 and 2950 Catalyst switches.



5.03. The CD includes a multimedia demonstration of using the history function on a Cisco router.

Editing the Command Line

The CLI editing features of the IOS are enabled by default. On the 1900 and 2950, you cannot disable the editing features; however, you can disable them on a Cisco router by using this command:

```
Router# terminal no editing
```

exam

Watch

Remember the basic editing control sequences for editing commands in the CLI. By default, the IOS stores the last ten executed commands.

To reenable the editing features, remove the **no** parameter from the preceding command (**terminal editing**).

Table 5-1 shows the control or command sequences that you can use to edit information in the CLI.



5.04. The CD includes a multimedia demonstration of using the command-line editing features on a Cisco router.

TABLE 5-1

Editing Control Sequences for IOS Devices

Control Sequence	Description
CTRL-A	Moves the cursor to the beginning of the line
CTRL-E	Moves the cursor to the end of the line
ESC-B	Moves the cursor back one word at a time
ESC-F	Moves the cursor forward one word at a time
CTRL-B	Moves the cursor back one character at a time
<LEFT ARROW>	Moves the cursor back one character at a time
CTRL-F	Moves the cursor forward one character at a time

TABLE 5-1

Editing Control Sequences for IOS Devices
(continued)

Control Sequence	Description
RIGHT ARROW	Moves the cursor forward one character at a time
CTRL-P	Recalls the last command
<UP ARROW>	Recalls the last command
CTRL-N	Recalls the most previously executed command
DOWN ARROW	Recalls the most previously executed command
CTRL-D	Deletes the character the cursor is under
<BACKSPACE>	Deletes the character preceding the cursor
CTRL-R	Redisplays the current line
CTRL-U	Erases the line completely
CTRL-W	Erases the word the cursor is under
CTRL-Z	Takes you from <i>Configuration</i> mode back to <i>Privilege EXEC</i> mode
<TAB>	Once you enter a few characters and hit the TAB key, the IOS device completes the word, assuming that you typed in enough characters to make the command or parameter unique
\$	When this appears at the beginning of a command line, it indicates that there are more characters to the right of the \$.

IOS Feature Example

Let’s use the **clock** command to illustrate the helpfulness of some of the IOS’s command-line features. This command is used on a router to set the current date and time. As an example, let’s assume that English isn’t your native language and that you are not sure how to spell “clock,” but you do know that it begins with the letters “cl.” Here’s an example:

```
Router# cl?
clear    clock
Router# cl
```

Notice two things about the output in this example. First, two commands begin with **cl**: **clear** and **clock**. Second, notice the CLI after the help output—the IOS kept the **cl** on the command line. Some administrators like this feature and some hate it. I’m in the latter camp, since I commonly forget that the IOS device is doing this and I start typing from the beginning, like this:

```
Router# clclock
Translating "clclock"
% Unknown command or computer name, or unable to find computer address
Router#
```

If you haven't guessed, this is an invalid command. Now that we know how to spell "clock," if you don't know what parameter(s) to type after the **clock** command, use the context-sensitive help:

```
Router# clock ?
set Set the time and date
Router# clock
```

The first column is the name of the parameter, and the second column is description. In this case, the IOS wants the word **set**:

```
Router# clock set ?
hh:mm:ss Current Time
Router# clock set
```

The next parameter wants the current time. This is based on UTC and is in a 24-hour format. For example, 3 P.M. would be 15:00:00. Again, use the context-sensitive help to figure out if there are more parameters:

```
Router# clock set 15:00:00 ?
<1-31> Day of the month
MONTH Month of the year
Router# clock set 15:00:00
```

Whenever you see a range of numbers in angle brackets, you must choose a value in this range. If you see a parameter in all caps, like MONTH, you must supply a *name*. In our clocking example, the IOS wants the name of the month, such as "May". Again, using the context-sensitive help, the IOS wants the number of the year following the name of the month:

```
Router# clock set 15:00:00 23 May ?
<1993-2035> Year
Router# clock set 15:00:00 23 May
```

Again, use the context-sensitive help to see what's next:

```
Router# clock set 15:00:00 23 May 2004 ?
<cr>
Router# clock set 15:00:00 23 May 2004
Router#
```

If you see “<cr>,” this means that you can hit the ENTER key and the IOS device will accept the command. On an IOS router, use the **show clock** command to see your current time and date:

```
Router# show clock
15:00:02.187 UTC Fri May 23 2003
Router#
```



5.05. The CD includes a multimedia demonstration of using a combination of the command-line editing features on a Cisco router.

EXERCISE 5-1



Using IOS Features

These last few sections have covered how you use the IOS features on your Cisco devices. Here are some exercises that you can perform on a Cisco router to enforce these skills. Use either the 2600 or 2500 router on the router simulator included on the CD-ROM, or you can use a real Cisco router. You can find a picture of the network diagram for the simulator in the Introduction to this book. Access the simulator and click on the *Lab Navigator* button. Double-click on *Exercise 5-1*, click on the *Load Lab* button, and then the OK button. Click on the *eRouters* button and choose 2600.

1. Access *User EXEC* mode on your router.
Hit the ENTER key. You should see the *EXEC* prompt: Router>.
2. Pull up the list of commands available at this mode.
Use the ?.
3. Go to *Privilege EXEC* mode.
Use the **enable** command and your prompt should look like this: Router#.
4. Type the **show interfaces** and **show running-config** commands on two separate command lines.
On one command line, type: **show interfaces**. Hit the ENTER key. On the next command-line type: **show running-config**.
5. Use the CLI editing features of your router by changing the **show running-config** command to **show startup-config** and execute this.
Use the command recall (UP ARROW) to recall the **show running-config** command. Edit this command and replace “running” with “startup.” You need to LEFT ARROW over to the “-” and BACKSPACE to delete the word “running.”

Then type **startup**. Use CTRL-E to go to the end of the line and hit the ENTER key to execute the command.

6. Log out of the router switch.
Use the **exit** command.

Now you should be more comfortable with the CLI of the IOS. The next section shows you how to create a basic configuration on your 1900 and 2950 switch.

Basic Switch Configuration

This section covers the basics of accessing the 1900 and 2950 Catalyst switches, creating a simple configuration on them, and using simple **show** commands.

Accessing the CLI

Accessing the CLI on the 1900 is different from accessing the CLI on the 2950. With the 1900, once the switch runs its hardware tests and loads the IOS, the IOS displays this menu:

```

-----
Catalyst 1900 Management Console
Copyright (c) Cisco Systems, Inc. 1993-1998
All rights reserved.
Enterprise Edition Software
Ethernet address:      00-C0-1D-81-A3-65

PCA Number:           73-3121-02
PCA Serial Number:    FAA0252A7RT
Model Number:         WS-C1924-EN
System Serial Number: FAA0304S0T5
Power Supple S/N:     PHI025178F2
-----

1 user(s) now active on Management Console.

                User Interface Menu
[M] Menus
[K] Command Line
[I] IP Configuration
[P] Console Password

Enter Selection:
```

If there is a password configured on the switch, you will be prompted for it before this menu is displayed. To choose an option, just type the letter. There are four options from this menu:

- **M** Use the menus to configure the switch.
- **K** Use the IOS CLI to configure the switch, which takes you to *User EXEC* mode.

exam

Watch

*For exam purposes, you only have to know how to use the IOS CLI (not the menu structure) on the 1900. Type **K** from the 1900's main menu to access the IOS CLI.*

- **I** Use a menu to create a basic IP configuration on the switch (this appears only if the switch has no IP addressing configured on it).
- **P** Assign a password to the console (this appears only if there is no console password configured).



5.06. The CD includes a multimedia demonstration of accessing the User EXEC CLI of the 1900.

EXEC Modes

On a 1900, once you type **K** from the main menu, you are taken into *User EXEC* mode. On the 2950, there is no menu system. When you boot up the 2950 switch, you are taken directly into *User EXEC* mode. Once there, use the **enable** command to go to *Privilege EXEC* mode:

```
> enable
#
```

Remember that the switches support context-sensitive help, so don't hesitate to take advantage of this powerful feature.

Accessing Configuration Mode

All changes on the 1900 and 2950 must occur within *Configuration* mode. To access this mode, you must first be at *Privilege EXEC* mode and use this command:

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

Notice that the prompt changed from “#” to “(config)#,” indicating the change in modes. You can abbreviate **configure terminal** to **conf t**. *Configuration* mode allows you to execute commands that change your switch’s configuration; however, you cannot actually view the changes from within this mode. To manage your switch, you’ll have to go back to *Privilege EXEC* mode by either typing **end** or pressing the control sequence CTRL-Z.

Assigning a Hostname

One of your first tasks is to change the name of your switch. This has only local significance and is used for management purposes. For instance, the Cisco Discovery Protocol (CDP) uses the hostname. CDP is discussed in Chapter 6. On both models of switches, the **hostname** command is used to change the name. Here is a simple example of changing the name on a 1900:

```
(config)# hostname 1900
1900(config)#
```

First, notice that you place the name of the switch after the command. Second, as soon as you hit ENTER, the new CLI prompt is different—it contains the switch’s new name.



5.07. The CD includes a multimedia demonstration of changing a 1900’s hostname.

In order to undo changes or negate a command on a switch, typically you precede the command with the **no** parameter. As an example, to change the 1900’s hostname back to the factory default, use this command:

```
1900(config)# no hostname 1900
(config)#
```

In certain cases, you don’t have to include the parameters of the command. In the preceding example, you could easily have typed in **no hostname** to accomplish the same thing.

exam

Watch

Use the *configure terminal* command to access Configuration mode. Use the *hostname*

command to change the name of the device—this name has local significance only.

Assigning Passwords

Both the 1900 and the 2950 allow you to set up passwords to restrict access to both *User* and *Privilege EXEC* modes. However, the commands to configure these passwords are *different* on each switch. The following two sections show you how to configure the *EXEC* passwords on your switches.

1900 Password Configuration The 1900 uses the same command to set both the *User* and *Privilege EXEC* passwords: **enable password level**. Here is the format of this command:

```
(config)# enable password level level_# password
```

exam

Watch

Use the *enable password level 1/15* command to configure the *EXEC* passwords on the 1900.

The level number is either **1** for *User EXEC* or **15** for *Privilege EXEC*. This is followed by the password. Please note that on the 1900 switch, the password is *case-insensitive*. To check your password configuration, log out of the switch and log back in: you should be prompted for your newly assigned passwords.



5.08. The CD includes a multimedia demonstration of configuring passwords on a 1900 switch.

2950 Password Configuration Configuring passwords on a 2950 switch is the same as configuring passwords on an IOS router (which is discussed later in this chapter). Unlike the 1900, which has one command to create your passwords, on the 2950, there are two commands.

The first major difference between the 1900 and 2950 *User EXEC* password configurations is that on the 1900, you used a single command to configure your *User EXEC* password, and this password is used to secure access to *only* the IOS CLI (not the console port of the switch and not telnet access). On the 2950, you can secure both console and telnet access. In order to do this, you must first go into the console interface or the telnet interface on your 2950. On the 2950, the **line console 0** command is used to do this:

```
Switch(config)# line console 0
Switch(config-line)# password console_password
```

The **0** in the first command specifies the console port. Lines and interfaces are numbered from 0 upward. Even though the 2950 has only a single console port, it is designated as 0. Next, notice that the prompt on the second line changed. *Configuration* mode actually has two different levels: *Global Configuration* and *Subconfiguration*. Certain commands on IOS devices take you into a specific *Subconfiguration* mode. Table 5-2 shows some configuration modes that you might see on IOS devices.

Please note that not all *Subconfiguration* modes are supported on all IOS devices. When you are in a *Subconfiguration* mode, the commands that you enter affect only that specific component of the router or switch. To leave a *Subconfiguration* mode and return to *Global Configuration* mode, use the **exit** command. Using the **end** command or CTRL-Z will always take you back to *Privilege EXEC* mode no matter what *Configuration* mode you are currently in. Here is an example:

```
Switch(config)# line console 0
Switch(config-line)# exit
Switch(config)#
```



5.09. The CD includes a multimedia demonstration of using Configuration and Subconfiguration modes on a 2950 switch.

In many cases, when you are in a *Subconfiguration* mode and type in a *Global Configuration* mode command, the IOS executes it and places you in *Global Configuration* mode, like this:

```
Switch(config)# line console 0
Switch(config-line)# hostname 2950
2950(config)#
```

Notice that when the **hostname** command was executed in *Line Subconfiguration* mode, the switch changed its name as well as the mode.

TABLE 5-2

Subconfiguration Modes on IOS Devices

IOS Prompt	Subconfiguration mode
(config-if)#	Interface Subconfiguration mode
(config-subif)#	Subinterface Subconfiguration mode
(config-line)#	Line Subconfiguration mode
(config-controller)#	Controller Subconfiguration mode
(config-router)#	IP Routing Protocol Subconfiguration mode

exam

Watch

Remember that Global Configuration mode commands can typically be executed at any Configuration level. These commands affect the IOS device as a whole, like the `hostname` command.

Subconfiguration mode commands only affect one specific component of an IOS device. Use the `exit` command to go back one Configuration level. Use the `end` or `CNTRL-Z` control sequence to exit Configuration mode.

Once you are in *Line Subconfiguration* mode, you can use the **password** command to assign the console password. Unlike the 1900, the password on the 2950 is case-sensitive. Remember that the **password** command, when executed under **line console 0**, sets the *User EXEC* password only for someone trying to access the 2950 from the console port only. Someone telnetting into the 2950 would not be prompted for a password. To set up a telnet password, use this configuration:

```
Switch(config)# line vty 0 15
Switch(config-line)# password telnet_password
Switch(config-line)# login
```

The **vtty** parameter in the preceding command refers to *virtual terminal*, a fancy name for telnet. The 2950 supports up to 16 simultaneous telnet connections, where each connection is internally tracked by a number: 0–15. You could assign a different password to each VTY, but then you really wouldn't know which password to use when telnetting into the switch. However, the IOS allows you to specify all 16 VTYS with the **line** command, simplifying your configuration.



5.10. The CD includes a multimedia demonstration of configuring passwords on a 2950 switch.

Once you are in *Line Subconfiguration* mode, use the **password** command to set your password. You also need to enter the **login** command to allow telnet access to the switch—this tells the IOS to use the password configured with the **password** command. There are other ways of checking access, such as a local username database or through an authentication server, but these concepts are beyond the scope of this book.

The second major difference between setting up passwords on the 1900 and the 2950 relates to configuring the *Privilege EXEC* password. Actually, the configuration is very similar between the two switches; however, the 2950 supports two commands for configuring this password:

```
Switch(config)# enable password Privilege_EXEC_password
-and/or-
Switch(config)# enable secret Privilege_EXEC_password
```

Both of these commands configure the *Privilege EXEC* password. The main difference is that using the **secret** parameter tells the 2950 to encrypt the password when it is saved, and using the **password** parameter doesn't. This is also true of the *Line Subconfiguration* mode **password** command. Passwords that are not encrypted can be encrypted by using the **service password-encryption** *Global Configuration* mode command. However, the **enable secret** command's encryption is much stronger than using the **service password-encryption** command. If you configure both the **enable password** and **enable secret** commands, the 2950 uses the password configured by the **enable secret** command to verify access to *Privilege EXEC* mode.

on the
Job

I recommend against using the `enable password` command along with `service password-encryption`, since there are utilities on the Internet, including Cisco's site, that can easily break this encryption. To my knowledge, there has been no report of anyone decrypting the password set with the `enable secret` command.

exam

Watch

You configure passwords on the 2950 as you do on the routers. Use the `password` command to secure line access and the `enable password` or `enable secret` command to secure

Privilege EXEC access. Remember that the `enable secret` command encrypts the password. With VTY access, you must also specify the `login` command within *Line Subconfiguration* mode.

Accessing and Configuring Interfaces

To configure an interface, you must first enter *Interface Subconfiguration* mode:

```
Switch(config)# interface ethernet|fastethernet|gigabitethernet
                    slot_#/port_#
Switch(config-if)#
```

You must specify two components to the **interface** command: the type and the location. On the 1900 switches, only Ethernet and Faster Ethernet interfaces are supported. On the 2950, the interfaces are Fast Ethernet and Gigabit Ethernet. Notice that the prompt changes, signifying that you are in *Interface Subconfiguration* mode.

on the
! o b

You can't tell what interface that you are actually in by examining the prompt. If you aren't sure, use the history recall feature to recall the interface command and reexecute it.

Following the type is the location, which is specified by the slot number, a slash (/), and the port. On the 1900, there are no modular slots; therefore, all fixed ports are considered to be in slot 0. Port numbers begin with 1 and work there way up to 27. Numbers 1–24 are the Ethernet ports, 25 is the AUI port, and 26 and 27 are the Fast Ethernet uplink ports. The 2950 fixed ports are considered to be in slot 0. The port numbers on the 2950 start at 1: 1–24 for a 2950-24 switch.

When specifying the interface, you can use any of the following:

```
(config)# interface ethernet 0/1
(config)# interface ethernet0/1
(config)# int e 0/1
(config)# int e0/1
```

e x a m

W a t c h

Remember how to enable and disable an interface on an IOS device: no shutdown and shutdown, respectively.

You can separate the type and location with a space, or concatenate the two together. Likewise, you can abbreviate the commands and parameters.

On the Catalyst switches, the interfaces are *enabled* by default. You can disable interfaces, though, with the **shutdown Interface Subconfiguration** mode command:

```
Switch(config)# interface type slot_#/port_#
Switch(config-if)# shutdown
```

To reenab the interface, use the **no shutdown** command.

Assigning IP Addressing Information

If you want to manage your 1900 or 2950 switch remotely, you need to assign it IP addressing information. For example, if you want to telnet to the switch, remotely manage it from a web browser or SNMP management station, or back up and restore configuration files or upgrade the switch, you'll need to set up IP addressing information on it. Each of the switches has different commands to assign this information. The next two sections cover the assignment of addressing information.

1900 IP Addressing One of the hardest concepts to grasp as a novice networker is IP addressing and how you perform subnetting. However, once you understand how IP addressing works, assigning IP addresses to your switches and routers is a simple process. Even though the 1900 has many interfaces, you can assign only a *single* IP address to the 1900 switch. This IP address is used only for management purposes and has nothing to do with how the 1900 switches frames between its interfaces.

Use the two following commands to set up basic IP connectivity:

```
(config)# ip address IP_address subnet_mask
(config)# ip default-gateway router's_IP_address
```

The **ip address** command assigns an IP address to the switch. Note that you must give both the IP address and the subnet mask in a dotted decimal format, like 192.168.1.5 255.255.255.0. By default, the IP address is placed in VLAN 1. (VLANs are covered in Chapter 8.) The **ip default-gateway** command tells the switch which router to use when the switch needs to reach a destination that is not in its configured subnet.



5.11. The CD includes a multimedia demonstration of configuring IP addressing information on a 1900 switch.

2950 IP Addressing On the 2950, IP addresses are configured differently than on the 1900:

```
Switch(config)# interface vlan1
Switch(config-vlan)# ip address IP_address subnet_mask
Switch(config-vlan)# exit
Switch(config)# ip default-gateway router's_IP_address
```

Unlike when working with the 1900, you must go into the VLAN interface that you want the IP address to be associated with. In most cases, this will be VLAN 1.

exam

Watch

The same commands are used to configure IP addresses on the switches: `ip address` and `ip`

`default-gateway`. The difference is that on the 2950, the address is configured under the VLAN interface.

(VLANs are covered in Chapter 8.) Once in the VLAN interface, use the **ip address** command to assign the address and mask. The 1900 and the 2950 use the same command to assign the default gateway: **ip default-gateway**.



5.12. The CD includes a multimedia demonstration of configuring IP addressing information on a 2950 switch.

Configuration Files

Configuration files can reside in many locations, including RAM, NVRAM, or a TFTP server, among other places. Whenever you make changes to a switch's configuration, these changes are made in RAM. These changes can then be saved to NVRAM (nonvolatile RAM), which is a static form of memory where, when the Cisco device is turned off, the contents are not erased and are available upon a power-up. The 1900 *automatically* copies any changes you make to NVRAM; the 2950, however, requires you to do this *manually*. Configuration changes can also be saved to a TFTP server—this, however, requires that you configure IP on your switch.

Viewing Your Configuration

To examine the active configuration on a 1900 or 2950 switch, use the **show running-config** command. You must be at *Privilege EXEC* mode to execute this command. Here is an example of this command from a 1900 switch:

```
# show running-config

Building configuration...
Current configuration:
!
interface Ethernet 0/1
!
interface Ethernet 0/2
!
<--output omitted-->
```

```

!
interface FastEthernet 0/27
!
line console
end
#

```

Saving Your Configuration

Configuration files are stored in NVRAM on both switches. When a switch boots up, the IOS then loads this configuration from NVRAM and places it in RAM. On the 1900 switch, you do *not* have to do anything to save the active, or running, configuration—within 30 seconds of executing a command, the 1900 automatically saves the configuration to NVRAM (therefore, wait at least this length of time before turning off the 1900).



5.13. The CD includes a multimedia demonstration of configuring on manipulating configuration files on a 2950 switch.

This is *not* true on the 2950 switch or IOS routers. Instead, you must execute the **copy running-config startup-config** *Privilege EXEC* mode command to save your changes. Upon executing this command, the 2950 or router takes the active configuration in RAM and saves it to NVRAM. In this process, the old configuration file in NVRAM is overwritten. Here is an example of using this command:

```

Switch# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Switch#

```

When executing this command, you are asked for a filename for the configuration file—the default is “startup-config.” This is the filename the IOS looks for when booting up. You can change the name for backup revisioning purposes (different versions of the backed-up configuration), but make sure that your most current configuration

exam

Watch

The 1900's configuration is automatically saved. On the 2950, use the *copy running-config startup-config* command to

save the configuration. The *show running-config* command displays the switch's currently running configuration in RAM.

is saved as “startup-config.” On the 2950, you can view the saved configuration in NVRAM with the **show startup-config** *Privilege EXEC* mode command.

Verifying Switch Operation

Besides using the **show running-config** command to verify your switch’s configuration, you can use many other commands. This section covers the **show interfaces**, **show ip**, and **show version** commands. All of these commands can be executed at either *User* or *Privilege EXEC* mode.

The show interfaces Command

To view the interfaces, status, and statistics for an interface, use the **show interfaces** command:

```
> show interfaces [type slot_#/port_#]
```

If you don’t list a specific interface, all of the interfaces on the switch are listed. Here is an example of this command on a 1900 switch:

```
> show interfaces ethernet 0/1
Ethernet 0/1 is Suspended-no-linkbeat
Hardware is Built-in 10Base-T
Address is 00E0.1EA1.a123
MTU 1500 bytes, BW 10000 Kbits
802.1d STP State: Blocking      Forward Transitions: 2
Port monitoring: Disabled
Unknown unicast flooding: Disabled
Unregistered multicast flooding: Disabled
Description:
Duplex setting: Full duplex
Back pressure: Disabled
```

Receive Statistics		Transmit Statistics	
-----		-----	
Total good frames	0	Total frames	0
Total octets	0	Total octets	0
Broadcast/multicast frames	0	Broadcast/multicast frames	0
Broadcast/multicast octets	0	Broadcast/multicast octets	0
Good frames forwarded	0	Deferrals	0
Frames filtered	0	Single collisions	0
Runt frames	0	Multiple collisions	0
No buffer discards	0	Excessive collisions	0
		Queue full discards	0
Errors:		Errors:	
FCS errors	0	Late collisions	0

Alignment errors	0	Excessive deferrals	0
Giant frames	0	Jabber errors	0
Address violations	0	Other transmit errors	0

Table 5-3 explains the status of the interface.

Table 5-4 explains some of the errors shown in the preceding 1900 code listing.



5.14. The CD includes a multimedia demonstration of displaying interface statistics on a 1900 switch.

Here is an example of the **show interfaces** command on a 2950 switch:

```
Switch# show interfaces fastethernet 0/1
FastEthernet0/1 is up, line protocol is up
  Hardware is Fast Ethernet, address is 0005.7428.1234
                                     (bia 0005.7428.1234)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  input flow-control is off, output flow-control is off
  Last input never, output 4d21h, output hang never
  Last clearing of "show interface" counters never
  Input queue:0/75/0/0 (size/max/drops/flushes);
                                     Total output drops:0

  Queueing strategy:fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    1 packets input, 64 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
<--output omitted-->
```

One of the first things that you want to examine in this display is the status of the interface: `FastEthernet0/1 is up, line protocol is up`. The first “up” refers to the status of the physical layer, and the second “up” refers to the status of the data link layer. Here are the possible values for the physical layer status:

- **Up** The switch is sensing a physical layer signal on the interface.
- **Down** The switch is not sensing a physical layer signal on the interface, a condition that can arise if the attached device is turned off, there is no cable attached, or you are using the wrong type of cable.
- **Administratively down** You used the **shutdown** command to disable the interface.

TABLE 5-3

Possible Statuses
for a 1900
Interface

Interface Status	Description
Enabled	The interface is functioning and can send and receive.
Disabled-mgmt	The interface has been disabled with the shutdown command.
Suspended-linkbeat	The switch cannot detect a physical layer signal on the interface.
Suspended-jabber	The attached device is causing excessive jabbering.
Suspended-violation	The interface is suspended because an address violation occurred.
Suspended-Spanning-Tree Protocol	Spanning Tree is currently running on the port and the port is not forwarding traffic.
Disabled-self-test	The interface has failed a hardware test.
Disabled-violation	The interface has been disabled because of an address violation and must be reenabled with the no shutdown command.
Reset	The interface is being reinitialized.

TABLE 5-4

Error Fields in
the 1900 **show
interfaces**
Command

Field Name	Description
FCS	Frame has the correct number of bytes, but an invalid FCS (frame check sequence—checksum).
Alignment	The frame has an incorrect number of bytes and an invalid FCS.
Giant Frames	The frame exceeds the maximum valid size.
Address Violations	The number of times an address violation occurred on the interface.
Late Collisions	The number of times a collision occurred after reading the first 512 bytes of a frame.
Excessive Deferrals	The number of times a excessive collisions occurred, possibly causing frames to be dropped.
Jabber Errors	The number of times a frame was received on the interface that exceed a certain time period.

Here are the possible values for the data link layer status:

- **Up** The data link layer is operational.
- **Down** The data link layer is not operational, a condition that can be caused by missed keepalives on a serial link, no clocking, or an incorrect encapsulation type.

exam

Watch

If the interface status is up and up, the interface is operational; if it is up and down, there is a problem with the data link layer; if it is down

and down, there is a physical layer problem; if it is administratively down and down, the interface was disabled with the `shutdown` command.



5.15. The CD includes a multimedia demonstration of displaying interface statistics on a 2950 switch.

The `show ip` Command

The **show ip** command is available only on the 1900 switch. This command displays the IP configuration of the switch:

```
> show ip
IP Address:192.168.1.12
Subnet Mask:255.255.255.0
Default Gateway:192.168.1.1
Management VLAN: 1
Domain name: dealgroup.com
Name server 1:192.168.1.3
Name server 2:192.168.1.3
HTTP server :Disabled
HTTP port : 80
RIP :Enabled
```



5.16. The CD includes a multimedia demonstration of displaying the IP configuration on a 1900 switch.

The `show version` Command

All IOS devices support the `show version` command. This command allows you to see what the model of the device is, the software running on it, how long it has been up, and its hardware characteristics, including the number and type of interfaces. Here is an example of this command on a 1900 switch:

```
> show version
Cisco Catalyst 1900/2820 Enterprise Edition Software
Version V9.00.00(12) written from 172.16.1.1
Copyright (c) Cisco Systems, Inc. 1993-1999
DS2820-1 uptime is 0day(s) 0hour(s) 8minute(s) 27second(s)
cisco Catalyst 1924 (486sxl) processor with 2048K/1024K bytes of memory
Hardware board revision is 5
Upgrade Status: No upgrade currently in progress.
Config File Status: No configuration upload/download is in progress
27 Fixed Ethernet/IEEE 802.3 interface(s)
Base Ethernet Address: 00-C0-1D-81-A3-65
```

In this example, you can see that this switch is running software version 9.00.00(12) and that it has 27 Ethernet interfaces.



5.17. The CD includes a multimedia demonstration of displaying software and hardware information on a 1900 switch.

EXERCISE 5-2



Configuring the 1900 Switch

These last few sections have covered how you create a basic configuration for a 1900 switch. Here are some exercises that you can perform on a 1900 switch to enforce these skills. If you are using the Boson's NetSim™ simulator on the CD-ROM, use the 1900-1 switch, which is a 1912. Otherwise, you can use a real 1900-series switch with the Enterprise Edition software. You can find a picture of the network diagram for the simulator in the Introduction to this book.

1. Access *User EXEC* mode on your 1900.

If you are using a real 1900-series switch, access the menus and type **K** to gain access to *User EXEC* mode; if you are using the simulator, start up the simulator.

Click on the *LabNavigator* button. Next, double-click on *Exercise 5-2* and click on the *Load Lab* button. At the top of the application in the menu bar, click on the *eSwitches* icon and choose *1900-1*. Next, hit ENTER to access *User EXEC* mode. You should see the *EXEC* prompt: >.

2. Access *Privilege EXEC* mode and then *Configuration* mode.

Use **enable** to go to *Privilege EXEC* mode and then **configure terminal** to access *Configuration* mode. Your prompt should look like this: (config) #.

3. Assign a hostname of *1900-1* to the 1900 switch.

Use the **hostname 1900-1** command and examine the prompt. Your prompt should look like this: 1900-1 (config) #.

4. Assign an IP address of 192.168.1.5/24 to the 1900, with a default gateway of 192.168.1.1.

Enter **ip address 192.168.1.5 255.255.255.0** and **ip default-gateway 192.168.1.1**.

5. Examine your active configuration.

Return to *Privilege EXEC* mode with the **end** command. Use the **show running-config** command.

6. Access *Host1* and change its address to 192.168.1.10/24. Also set the default gateway to 192.168.1.1. Test the connection by pinging the 1900-1 switch.

Click the *eStations* icon in the toolbar of the simulator and select *Host1*. On the CLI, type **ipconfig /ip 192.168.1.10 255.255.255.0**. Then, type **ipconfig /dg 192.168.1.1**. Test the connection: **ping 192.168.1.5**.

7. Go back to *Privilege EXEC* mode on the 1900-1 switch. Ping *Host1*.

Disable interface **ethernet 0/1**. Trying pinging again. What happens? Enable interface *e0/1*. Ping *Host 1* again.

Click the *eSwitches* icon in the toolbar of the simulator and select *1900-1*. Ping *Host 1*: **ping 192.168.1.10**. Go into *Configuration* mode: **configure terminal**. Disable interface *e0/1*: **configure terminal, interface ethernet 0/1, shutdown, end**. Ping again: **ping 192.168.1.10**. The ping should fail. Reenable interface *e0/1*: **configure terminal, interface ethernet 0/1, shutdown, end**. Ping again: **ping 192.168.1.10**. The ping test should be successful.

Now you should be more comfortable with the basic configuration of the 1900 switch. You'll build upon this configuration throughout the rest of this chapter. Don't close the simulator if you wish to complete Exercise 5-3 and 5-4!

EXERCISE 5-3



Configuring the 2950 Switches

In the last exercise, you created a basic configuration on a 1900 switch. In this exercise, you will create a basic configuration on a 2950 switch. If you are using Boson's NetSim™ simulator on the CD-ROM, use the 2950-1 switch. If you have closed the simulator since the last lab, you will need to complete Exercise 5-2 first. At the top of the application in the menu bar, click on the *eSwitches* icon and choose 2950-2. You can find a picture of the network diagram for the simulator in the Introduction to this book.

1. Go to *Configuration* mode on your 2950-1 switch.
Access *User EXEC* mode. Use **enable** to go to *Privilege EXEC* mode and then **configure terminal** to access *Configuration* mode. Your prompt should look like this: `Switch(config)#`.
2. Assign a hostname of 2950 to the switch of 2950-1.
Use the **hostname 2950-1** command and examine the prompt.
3. Assign an encrypted *Privilege EXEC* password of *cisco*.
Enter **enable secret cisco**.
4. Assign an IP address of 192.168.1.2/24 to the 2950 in VLAN 1, with a default gateway of 192.168.1.1.
Enter the VLAN interface with **interface vlan1**. Next, enter the addressing information: **ip address 192.168.1.4 255.255.255.0**. Exit the interface with the **exit** command and configure the default gateway: **ip default-gateway 192.168.1.1**.
5. Save your configuration to NVRAM and view the configuration in NVRAM. Test connectivity by pinging the 1900-1 switch.
Return to *Privilege EXEC* mode with the **end** command. Save it with **copy running-config startup-config** and view it with **show startup-config**. Test connectivity: **ping 192.168.1.5**. The ping should be successful.

6. Configure an IP address of 192.168.1.11 on *Host4* and a default gateway of 192.168.1.1. Test this by pinging *Host1* and 1900-1.
Click the *eStations* icon in the tool bar and select *Host4*. On the CLI, type **ipconfig /ip 192.168.1.11 255.255.255.0**. Then, type **ipconfig /dg 192.168.1.1**. Test the connection: **ping 192.168.1.10** and **ping 192.168.1.5**.
7. Configure the 2950-2 switch. The commands are the same, just use the appropriate configuration information: the hostname is 2950-2 and the IP address is 192.168.1.3/24. Test connectivity to the 1900-1 and 2950-1 switches.
Click the *eSwitches* icon in the tool bar and select 2950-2. On the 2950-2 switch, access *User EXEC* mode, then enter: **enable, configure terminal, hostname 2950-2, enable secret cisco, interface vlan1, ip address 192.168.1.3 255.255.255.0, exit, ip default-gateway 192.168.1.1, end, copy running-config startup-config**, and **show startup-config**. Make sure you configured the right hostname and IP address. Test connectivity by pinging the 1900-1 and 2950-1 switches: **ping 192.168.1.5** and **ping 192.168.1.4**. The pings should be successful.

Now you should be more comfortable with the basic configuration of the 2950 switch. In the next section, you will be shown how to create a basic configuration on your Cisco router. You'll build upon the configuration you built in Exercise 5-2 and 5-3 throughout the rest of this chapter, so don't close the simulator yet!

Basic Router Configuration

This section covers some of the basic commands that you can use to access and configure a Cisco IOS router, which are, as you will see, much like those on the 2950 Catalyst switch. The advantage this provides is that the you don't have to learn a complete new CLI. The first thing that is covered is the *System Configuration Dialog* script, which prompts you for information about how you want to configure your router. I'll then cover the commands used to create a very basic configuration on your IOS router.

Setup Mode

When a router boots up, runs its hardware diagnostics, and loads the IOS software, the IOS then attempts to find a configuration file in NVRAM. If it can't find a configuration

file to load, the IOS then runs the *System Configuration Dialog*, commonly referred to as *Setup* mode, which is a script that prompts you for configuration information. The purpose of this script is to ask you questions that will allow you to set up a basic configuration on your router: It is not intended as a full-functioning configuration tool. In other words, the script doesn't have the ability to perform all the router's configuration tasks. Instead, it is used by novices who are not that comfortable with the IOS CLI. Once you become familiar with the CLI and many of the commands on the router, you'll probably never use this script again in your life.

Running the System Configuration Dialog

As was mentioned in the last paragraph, one way to access the *System Configuration Dialog* is to boot up a router without a configuration in NVRAM. The second way is to use the **setup** *Privilege EXEC* mode command, shown here:

```
Router# setup

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: y
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

First, would you like to see the current interface summary? [yes]:

Interface IP-Address  OK? Method Status          Protocol
BRI0      unassigned  YES unset  administratively down down
BRI0:1    unassigned  YES unset  administratively down down
BRI0:2    unassigned  YES unset  administratively down down
Ethernet0 unassigned  YES unset  administratively down down
Serial0   unassigned  YES unset  administratively down down

Would you like to enter basic management setup? [yes/no]: n
Configuring global parameters:
  Enter hostname [Router]:
  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: dealgroup1
  The enable password is used when you do not specify an
  enable secret password, with some older software versions,
```

```

and some boot images.
Enter enable password: dealgroup2
The virtual terminal password is used to protect
access to the router over a network interface.
Enter virtual terminal password: cisco
Configure SNMP Network Management? [no]:
Configure LAT? [yes]: n
Configure AppleTalk? [no]:
Configure DECnet? [no]:
Configure IP? [yes]:
Configure IGRP routing? [yes]: n
Configure RIP routing? [no]:
<--output omitted-->
BRI interface needs isdn switch-type to be configured
Valid switch types are :
    [0] none.....Only if you don't want to
        configure BRI.
    [1] basic-1tr6....1TR6 switch type for Germany
    [2] basic-5ess....AT&T 5ESS switch type for the
        US/Canada
    [3] basic-dms100..Northern DMS-100 switch type
<--output omitted-->
Choose ISDN BRI Switch Type [2]:

Configuring interface parameters:
Do you want to configure BRI0 (BRI d-channel) interface? [no]:
Do you want to configure Ethernet0 interface? [no]: y
Configure IP on this interface? [no]: y
IP address for this interface: 172.16.1.1
Subnet mask for this interface [255.255.0.0] : 255.255.255.0
Class B network is 172.16.0.0, 24 subnet bits; mask is /24
Do you want to configure Serial0 interface? [no]:
Do you want to configure BRI0 interface? [no]:

The following configuration command script was created:
hostname Router
enable secret 5 $1$/Cck$4r7zDwDNeqkxF0.kJxC3G0
enable password dealgroup2
line vty 0 4
  password cisco
no snmp-server
!
no appletalk routing
no decnet routing
ip routing
<--output omitted-->
end

```

[0] Go to the IOS command prompt without saving this config.

```
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.
Enter your selection [2]:
```

Information included in brackets ([]) are default values—if you just hit ENTER, the value in the brackets is used. One problem with the script is that if you make a mistake, there is no method of going back to the preceding question. Instead, you can use the CTRL-C break sequence to abort the script. The following sections break down the different components of the script.



The questions that the script asks you might be different from router to router, depending on the hardware model and the software running on it.

Status and Global Configuration Information

At the beginning of the script, you are asked whether or not you want to continue. If you answer “yes” or “y,” the script will continue; otherwise, if you answer “no” or “n,” the script is aborted and you are returned to *Privilege EXEC* mode. The second thing that you are asked is if you want to see the status of the router’s interfaces. If you answer “y,” then you’ll see all of the interfaces on the router, the interfaces’ IP addresses, and the status of the interfaces.

After the status information, you are taken into the actual configuration. The first part of the configuration deals with all configuration information for the router except for the interfaces, which is the second part. In this part of the configuration, you are asked for things like the *Privilege EXEC* password, VTY (telnet) password, which network protocols you want to activate, and other information.



Note that you are prompted for two Privilege EXEC passwords in the script: *enable secret* and *enable password*. These passwords are the same as discussed earlier in the 2950 configuration section. Even though you would normally configure only one, the script requires you to enter both; it also requires that both passwords be different. Just as with the 2950 configuration, the *enable secret* password takes precedence over the *enable password*.

Protocol and Interface Configuration Information

After configuring the global information for the router, you are then lead through questions about which interfaces you want to use and how they should be configured. The script is smart enough to ask only configuration questions based on how you answered the global questions. As an example, if you activate IP, the script asks you

for each activated interface, if you want the interface to process IP, and if yes, the IP addressing information for the interface. If you have an ISDN interface in your router, as in the preceding example, you will be prompted for the ISDN type of the switch to which you are connecting your router. ISDN is covered in Chapter 17.

Exiting Setup Mode

After you answer all of the script's configuration questions, you are then shown the router configuration the script created using the answers to the script's questions. Please note that the IOS hasn't yet activated the configuration file. Examine the configuration closely and then make one of the three choices shown in Table 5-5. Also, if you enter 1 as your option, when the script starts over again, the information that you previously entered appears in brackets and will be the default values when you hit the ENTER key on an empty line. Older versions of the IOS asked a simple yes/no question at the end as to whether or not you wanted to use the configuration you just created with the script utility.

exam

Watch

Remember that this script is started when the router boots up and there is no configuration in NVRAM or you use the *setup* command from

Privilege EXEC mode. Also, know the three options at the end of the Setup dialog script. You can use the CTRL-C break sequence to abort the script.

TABLE 5-5

Options at the End of the **System Configuration Dialog**

Option	Description
0	Discard the script's configuration and return to <i>Privilege EXEC</i> mode.
1	Return to the beginning of the script.
2	Activate the script's configuration, save the configuration to NVRAM, and return to <i>Privilege EXEC</i> mode.



The 2950 switch also supports a System Configuration Dialog; however, as it is a layer-2 device, the number of questions asked is much smaller during the configuration and is typically used only to configure passwords and an IP address on it.



5.18. The CD includes a multimedia demonstration of using the System Configuration Dialog on a Cisco router.

Configuration Mode

In most situations, you'll enter *Configuration* mode on the router and manually enter the commands. One of the advantages of this approach is that you have full access to every command that the router supports, whereas the *System Configuration Dialog* supports only a small subset of commands. Accessing *Configuration* mode on the router is the same as accessing it on a 1900 or 2950 switch:

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

Moving around *Configuration* mode is also the same, as well as exiting *Configuration* mode with the **end** command or the CTRL-Z control sequence. And like the switches, the router supports both *Global Configuration* and *Subconfiguration* modes. The following sections cover some of the basic commands for configuring an IOS-based router.

Assigning a Hostname

When you get a router from Cisco, the default name of the router is "Router." To change your router's name, use the **hostname** command. Here is an example of its use:

```
Router(config)# hostname bullmastiff
bullmastiff(config)#
```

As you can see, this is the same command used on both of the Catalyst switches. Again, notice that the prompt changed: It now includes the name of the router.

Configuring Passwords

Cisco routers support two levels of passwords: *User EXEC* and *Privilege EXEC*. Configuring these passwords is exactly the same as configuring them on the 2950 switch. The following sections cover the configuration of these passwords on a Cisco router.

User EXEC Passwords Configuring the router's *User EXEC* passwords is straightforward. For each type of access—console, telnet, or the auxiliary port—you

exam**Watch**

Remember how to configure passwords on routers, especially how to type them in.

can have the same or different passwords, since these passwords are configured under the appropriate line type. Here is the configuration for setting up *User EXEC* passwords:

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

```
Router(config-line)# exit
Router(config)# line vty 0 4
Router(config-line)# login
Router(config-line)# password telnet_password
```

As you can see, this is no different from configuring the *User EXEC* passwords on a 2950 switch. The only difference between the switch and the router is the number of VTYs supported for telnet access. Depending on the router model and IOS software version, this number might range from 5 (0-4) on up to almost 1000.

About the only other minor difference is that some routers have an auxiliary port. This is typically used as a backup console port, or a remote access port with a modem attached to it. The following code shows the syntax for setting up password authentication on it:

```
Router(config)# line aux 0
Router(config-line)# password console_password
Router(config-line)# exit
```

Privilege EXEC Passwords Configuring the *Privilege EXEC* password on the router is the same as on a 2950 switch:

```
Router(config)# enable password Privileged_EXEC_password
                    -and/or-
Router(config)# enable secret Privileged_EXEC_password
```

Remember that the **enable secret** command encrypts the password, while the **enable password** command doesn't.



5.19. The CD includes a multimedia demonstration of setting up passwords on a Cisco router.

Setting Up a Login Banner

You can set up a login banner on your router that will display a message to every user attempting to access *User EXEC* mode. The **banner motd** command is used to create the login banner:

42 Chapter 5: Basic Switch and Router Configuration

```
Router(config)# banner motd start_and_delimiting_character
enter_your_banner_here
enter_the_delimiting_character_to_end_the_banner"
Router(config)#
```

“MOTD” is a carryover from the Unix world that stands for “message of the day.” After the **banner motd** command, you must enter the starting character. This is the character that is used to signify the beginning and ending of the banner. Once you enter the starting character, when the IOS CLI parser sees this character later in your text, the IOS terminates the banner and returns you to the CLI prompt. One nice feature of the banner is that the ENTER key doesn’t terminate the banner, and so you can have banners that span multiple lines. Whenever you have completed your banner, type the ending character (the one that you began the banner with).

Here is an example of setting up a login banner:

```
Router(config)# banner motd $
This is a private system and only authorized individuals
are allowed!
All others will be prosecuted to the fullest extent of the law!
$
Router(config)#
```

In this example, the banner spans multiple lines and the delimiting character is the dollar sign (\$).

A banner doesn’t have to span multiple lines, though, but can be placed on a single line, as in this example:

```
Router(config)# banner motd 'Keep Out!'
Router(config)#
```

In this example, the single quote (‘) is the delimiting character. Once you have created a login banner, test it by logging out of the router and logging back in. You should see your banner appear before the *User EXEC* password prompt.



5.20. The CD includes a multimedia demonstration of setting up a login banner on a Cisco router.

e x a m

W a t c h Use the *banner motd* command to set up a login banner. Use the *exec-timeout* command to set up the EXEC timeout.

Changing the Inactivity Timeout

By default, the router automatically logs you off after ten minutes of inactivity. You can change this with the **exec-timeout** *Line Subconfiguration* mode command:

```
Router(config)# line line_type line_#
Router(config-line)# exec-timeout minutes seconds
```

If you don't want a particular line to ever time out, you can set the minutes and seconds values to zero, like this:

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



Disabling the timeout for a line is not recommended in a production environment, since it creates security issues. However, for training purposes, such as studying for the CCNA exam, this is okay.

To verify your line configuration, use the **show line** command:

```
Router# show line con 0
Tty Typ Tx/Rx A Modem Roty AccO AccI Uses Noise Overruns
* 0 CTY - - - - - 0 0 0/0

Line 0, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Status: Ready, Active
Capabilities: none
Modem state: Ready
Special Chars: Escape Hold Stop Start Disconnect Activation
                ^x none - - none
Timeouts: Idle EXEC Idle Session Modem Answer Session Dispatch
           never never none none not set
Session limit is not set.
Time since activation: 0:04:49
Editing is enabled.
History is enabled, history size is 10.
Full user help is disabled
Allowed transports are pad telnet mop. Preferred is telnet.
No output characters are padded
No special data dispatching characters
```

In this example, notice that the timeout value is now set to “never.”



5.21. The CD includes a multimedia demonstration of changing the idle timeout on a Cisco router.

CLI Output

One nice feature of an IOS device is that when certain types of events occur, such as an interface going down or up, an administrator making a configuration change, or output of **debug** commands, the router, by default, prints an informational message on the console port. It won't, however, display the same messages if you happen to have telnetted into the router or accessed it via the auxiliary port.

If this is the case, you can have the router display these messages on your screen by executing the *Privilege EXEC* **terminal monitor** command, as shown here:

```
Router# terminal monitor
```

You must execute this command once you have logged into the router from the VTY or auxiliary lines. This command is *not* saved to NVRAM when you execute the **copy running-config startup-config** command—it is used only during the active management session. Once you execute this command, the router displays the information messages on screen. Once you log out, however, this command does not apply to anyone else telnetting into the router. Each individual telnetting into the router must re-execute this command.

One annoying problem with the information messages, however, is that if you are typing in a router command, when the router displays the message, it starts printing it right where the cursor is, making it hard for you to figure out where you left off typing. If you remember, just keep typing your command; or hit CTRL-C to abort the

command. A better approach, though, is to set up the router so that after the message prints on your screen, the router redisplay what you have already typed in on a new prompt. The **logging synchronous** command accomplishes this:

exam

Watch Use the *terminal monitor* command to view console output on non-console lines.

```
Router(config)# line line_type line_#
Router(config-line)# logging synchronous
```

First, notice that this command is executed under the *Line Subconfiguration* mode. Next, if you want to implement this feature, you'll need to set it up under all the lines from which you'll be accessing your router, including your VTYS and console port. Note that you can also use the CTRL-R control sequence to refresh the screen.



5.22. The CD includes a multimedia demonstration of using logging synchronous on a Cisco router.

Configuring Router Interfaces

Accessing interfaces on IOS routers is the same as accessing interfaces on the 1900 and 2950 Catalyst switches. To access an interface and enter *Interface Subconfiguration* mode, use the **interface** command:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)#
```

Unlike the Catalyst switches, which support only Ethernet-type interfaces, Cisco routers support a wide variety of interfaces, including synchronous **serial** and **async** serial, ISDN **bri** and **pri**, **atm**, **fddi**, **tokenring**, **ethernet**, **fastethernet**, and **gigabitethernet**, as well as others. Of course, not every Cisco router supports all types of interfaces. As an example, routers in the 800 series support only serial, ISDN, and Ethernet.

After the type comes the location of the interface. Slot numbers begin with 0 and port numbers begin with 0. Therefore, if you had an interface like **ethernet 0/0**, this would be the first slot in the router, and the very first port; whereas, **ethernet 1/1** would be in the second slot, and the port would be the second port.

Some routers do not support slots or modules, and therefore this is omitted—instead you just list the port number. Examples of routers that do not have slots or slot numbers are the 800, 1600, 1700, and 2500 routers. Here are some examples of interface names where the router lacks slots:

- **ethernet 0** or **e0**
- **serial 0** or **s0**
- **bri 0**

For those routers that support slots, the 3600 and 7200 routers, for example, you must specify the slot number, followed by a slash, and then the port number. Here are some examples:

- **ethernet 0/0** or **e0/0**
- **serial 1/0** or **s1/0**

Remember that when listing the type and slot and port numbers, you *can* concatenate these values as shown in these examples.

Including an Interface Description You can add a description to any interface on a router, or for that matter a switch, by using the **description** command:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# description interface_description
```

The description is a one-line description describing the device that the interface is connected to, or whatever description you want to assign. This description appears in the output of the **show interfaces** command.

Enabling and Disabling Interfaces Unlike Catalyst switches, Cisco router interfaces are *disabled* by default. For each interface that you want to use, you must go into the interface with the **interface** command and activate it with the **no shutdown** command:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# no shutdown
```

Whenever the interface changes status, the router prints an information message on the screen. Here is an example of where an interface on a router is being activated:

```
Router(config)# interface fastethernet0
Router(config-if)# no shutdown
1w0d: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to up
1w0d: %LINEPROTO-5-UPDOWN: Line protocol on Interface
      FastEthernet0, changed state to up
Router(config-if)#
```

exam

Watch

When copying and pasting a configuration file into the router, and the router interface is disabled with the *shutdown* command, your pasted configuration file must contain the *no shutdown* command in order to active the interface. This is a common problem when copying and pasting a configuration file from an old router to a new router, where the interfaces on the new router are disabled by default.

In this example, the first information line indicates that the physical layer is activated. The second line indicates that the data link layer is enabled. If you want to disable a router's interface, enter *Interface Subconfiguration* mode for that particular interface and execute the **shutdown** command.



5.23. The CD includes a multimedia demonstration of enabling and disabling interfaces on a Cisco router.

Configuring LAN Interfaces Some routers, like the 4000 series, support dual Ethernet connectors for a single interface. With some of these routers, the IOS is intelligent enough to figure out which connector is being used and handles the configuration automatically. In other cases, though, you must tell the router which connector the interface should use by configuring the **media-type** command on the interface:

```
Router(config)# interface ethernet [slot_#/]port_#  
Router(config-if)# media-type media_type  
Router(config-if)# speed 10|100|auto  
Router(config-if)# [no] half-duplex
```

Here are the media types that you can specify: **au**i, **10baset**, **100baset**, and **mi**i.

For 10/100 Ethernet ports that support auto-sensing, it is recommended that you hard-code the speed and duplexing with the **speed** and **half-duplex** commands. Setting the **speed** to **auto** has the interface auto-sense both the speed and the duplexing. In order to hard-code the duplexing, you must first hard-code the speed to 10 or 100. To set the port to full-duplex, use the **no half-duplex** command.

Configuring Serial Interfaces When connecting a serial cable to the serial interface of the router, typically clocking is provided by an external device, such as a modem or a CSU/DSU. The router is the DTE and the external device is the DCE, where the DCE provides the clocking. This type of WAN connection is discussed further in Chapter 15.

In some cases, however, you might connect two routers back-to-back using the routers' serial interfaces. For instance, if you are building your own lab to practice CCNA commands, you'll more than likely connect the routers back-to-back to reduce equipment costs. In this situation, each router, by default, is a DTE. Since clocking is required in order for the interface to be enabled, one of the two routers will have to perform the function of an external DCE. This is accomplished by using the **clock rate** *Interface Subconfiguration* mode command on the a serial interface:

```
Router(config)# interface serial [slot_#/]port_#  
Router(config-if)# clock rate rate_in_bits_per_second
```

When entering the clock rate, you can't just choose any arbitrary value. Use the context-sensitive help to find out which clock rates your serial interface supports.

exam

Watch

Use the `show controller` command to determine DTE and DCE connections. Use the `clock rate` command to configure the speed for DCE connections.

Here are some possible values: **1200, 2400, 4800, 9600, 19200, 38400, 56000, 72000, 125000, 148000, 500000, 800000, 1000000, 1300000, 2000000, and 4000000.**

Please note that that you can't choose an arbitrary router in the back-to-back connection to be DCE—this is based on how the two routers are cabled. One end of the cable is physically the DTE, and the other is the DCE. Some cables are

marked and some are not, depending on where you purchased them from. If you are not sure which router has the DTE end of the cable and which one has the DCE end, you can determine this with the **show controller** command:

```
Router> show controller serial [slot_#/]port_#
```

This is one of the few commands where you *cannot* concatenate the type and the port number—you must separate them by a space. Here is an example of the use of this command:

```
Router> show controller serial 0
HD unit 0, idb = 0x121C04, driver structure at 0x127078
buffer size 1524 HD unit 0, DTE V.35 serial cable attached
<--output omitted-->
```

Notice that the second line of this example holds two important pieces of information: the connection type (DTE) and the type of cable (V.35). Here is an example of an interface connected to the end of a DCE cable:

```
Router> show controller serial 0
HD unit 0, idb = 0x1BA16C, driver structure at 0x1C04E0
buffer size 1524 HD unit 0, V.35 DCE cable, clockrate 64000
<--output truncated-->
```

In this example, the clocking has already been configured: 64,000 bps (bits per second).



5.24. The CD includes a multimedia demonstration of setting the clocking on a serial interface on a Cisco router.

Configuring the Bandwidth Parameter All interfaces have a bandwidth value assigned to them. This is used by certain routing protocols, such as IGRP,

OSPF, and EIGRP, when making routing decisions. (Routing protocols are covered in Chapters 9, 10, and 11.) For LAN-based interfaces, the speed of the interface becomes the bandwidth value, where the bandwidth is measured in kilobits per second (Kbps). However, on serial interfaces, the bandwidth defaults to 1,554 Kbps, or the speed of a T1 link. This is true no matter what the physical clock rate is on the interface (discussed in the preceding section). To change the bandwidth value for an interface, use the **bandwidth** *Interface Subconfiguration* mode command:

```
Router(config)# interface serial [slot_#/]port_#
Router(config-if)# bandwidth rate_in_Kbps
```

As an example, a serial interface clocked at 56,000 bps should have its bandwidth value changed to 56 Kbps, like this:

```
Router(config)# interface serial 0
Router(config-if)# bandwidth 56
```

exam

Watch

*Please note that the **bandwidth** command does not change the clock rate on an interface: the *clock**

*rate command does this. The **bandwidth** command affects only routing protocols that use bandwidth as a metric.*

Configuring IP Addressing Information

You can use many commands on the router to set up your IP addressing information. One of the most common is to assign an IP address to an interface; however, there are many more commands, including the setup of DNS, determining how subnet masks appear in the output of **show** commands, restricting directed broadcasts, and others. The following sections cover these configurations.

Assigning IP Addresses Unlike the 1900 and 2950, which only need a single IP address, routers need a unique IP address on each interface that will route IP traffic. Actually, each interface from a router is a separate network or subnet, and therefore you need to plan your IP addressing appropriately and assign a network number to each router segment and then take an unused host address from the segment and configure it on the interface of the router. IP addressing and its components is discussed in Chapter 3.

Let's look at a couple of examples of incorrectly and correctly assigning IP addresses to a router's interfaces. Figure 5-1 shows an invalid configuration example.

In this example, there is only one network number: 192.168.1.0/24. Notice that *each* interface on the router has an address from this network number. Actually, if you would try to configure this addressing scheme on a router, you would get an overlapping address error and be prevented from completing the addressing configuration.

Each interface needs a unique host address, as is shown in Figure 5-2. Notice that in this example, each interface has an address from a *different* network number when compared to the other interfaces on the router. As to which host address you choose for the router interface, this is up to personal preference. Many administrators either use the first or last host address in the network number for the router's interface, but any valid, unused host address from that network number can be used.

As you have probably already guessed, configuring an IP address on a router requires you to be in *Interface Subconfiguration* mode. Here is the syntax of this command:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# ip address IP_address subnet_mask
```

This syntax, as you can see, is the same as configuring an IP address on the 2950 switch. You can verify your IP addressing configuration with the **show interfaces** or **show ip interfaces** command, discussed later, in the section “Verifying a Router's Operation.”

Using the example in Figure 5-2, here would be the router's IP addressing configuration:

```
Router(config)# interface ethernet 0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface ethernet 1
Router(config-if)# ip address 192.168.2.1 255.255.255.0
Router(config-if)# no shutdown
<--output omitted-->
```

FIGURE 5-1
Invalid addressing configuration for a router

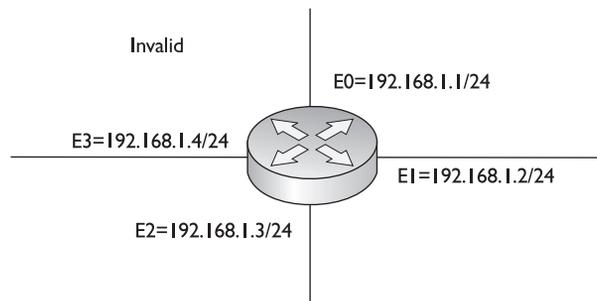
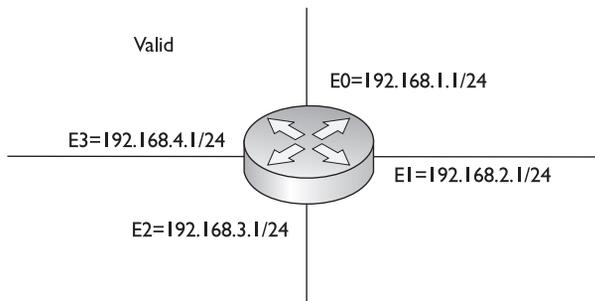


FIGURE 5-2

Correct
addressing
configuration
for a router



5.25. The CD includes a multimedia demonstration of configuring an IP address on an interface of a Cisco router.

exam

Watch

If you misconfigured an IP address on a router's interface, use the *no ip address* command to remove it. Optionally, you can use the *ip address*

command with the correct IP address and subnet mask to overwrite the existing configuration. Remember how to configure IP addressing on a router!

Subnet Zero Configuration Starting with IOS 12.0, Cisco automatically allows you to use IP subnet zero networks—the first network number in a subnetted network. Prior to IOS 12.0, you were not, by default, allowed to use these subnets. However, you could enable their use if you needed extra networks by configuring the **ip subnet-zero** command:

```
Router(config)# ip subnet-zero
```

Directed Broadcast Configuration In version 11.x and earlier of the IOS, the router would automatically forward directed broadcasts. As you may recall from Chapter 3, each network number has its own broadcast address. You can send packets with this address as a destination, which a router can forward to this destination segment so that all hosts on that segment receive it. However, many unscrupulous individuals have taken advantage of this process to flood a network segment with these broadcasts. Therefore, in IOS 12.0 and later, directed broadcasts are disabled on a router's interfaces, causing the router to drop any received directed broadcast packets.

If you want to reenable this function, use the following *Interface Subconfiguration* mode command:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# ip directed-broadcast
```

If you want to disable directed broadcasts again, just preface the preceding command with the **no** parameter. You can verify your directed broadcast configuration with the **show ip interfaces** command, discussed later, in the section “Verifying a Router’s Operation.”

IP Subnet Mask Display Configuration Whenever you use many **show** commands on your router and IP addresses are displayed, the default display format for subnet masks is dotted-decimal. You can change the display format of subnet masks with the following command:

```
Router# term ip netmask-format bit-count|decimal|hexadecimal
```

To begin with, execution of this command is *not* done from *Configuration* mode. Therefore, this change takes effect only for your current login session—as soon as you log out and log back in again, the default is to display the masking information in the dotted-decimal format. Here are your display options:

- **bit-count** Example: 192.168.1.0/24
- **decimal** Example: 192.168.1.0 255.255.255.0 (default format)
- **hexadecimal** Example: 192.168.1.0 0xFFFFF00

Of course, you might get tired of continually retyping this command every time you log in. Therefore, you can configure a default behavior that can be saved by the router and will be the same every time that you log into the router. This is done from *Line Subconfiguration* mode:

```
Router(config)# line line_type line_#
Router(config-line)# ip netmask-format bit-count|decimal|hexadecimal
```

Note that you will need to execute this command on each line from which you access your router, as the following configuration shows:

```
Router(config)# line console 0
Router(config-line)# ip netmask-format bit-count
Router(config-line)# exit
Router(config)# line vty 0 4
Router(config-line)# ip netmask-format bit-count
```



To verify your configuration, use the **show interfaces** or **show ip interfaces** command to verify that the subnet mask is displaying correctly. These commands are discussed later, in the section “Verifying a Router’s Operation.”

5.26. The CD includes a multimedia demonstration of changing the subnet mask display on a Cisco router.

Static Host Configuration As you are well aware, in the IP world, we typically don’t type in the IP address to reach a destination. For example, if you want to reach Cisco’s site, in your web browser address bar, you type **www.cisco.com** or **http://www.cisco.com**. Your web browser then resolves the host and domain names to an IP address. The router also supports hostnames for certain operations, such as ping and telnet, as is discussed in Chapter 6.

There are two basic ways to have your router resolve hostnames to IP addresses: static and dynamic (using DNS). You can create a static resolution table by using this command:

```
Router(config)# ip host name_of_host [TCP_port_#]
                    IP_address_of_host [2nd_IP_address...]
```

You must first specify the name of the remote host. Optionally, you can specify a port number for the host—this defaults to 23 for telnet if you omit it. After this, you can list up to eight IP addresses for this host. The router will try to reach the host with the first address, and if that fails, try the second address, and so on and so forth. Use the **show hosts** command to examine your static entries. This command is covered later, in the section “Verifying a Router’s Operation.”

DNS Resolution Configuration If you have access to a DNS server or servers, you can have your router use these to resolve names to IP addresses. This is configured with the **ip name-server** command:

```
Router(config)# ip name-server IP_address_of_DNS_server
                    [2nd_server's_IP address ...]
```

You can list up to six DNS servers for the router to use with this command. Use the **show hosts** command to examine your static and dynamic entries. This command is covered later, in the section “Verifying a Router’s Operation.”

Many administrators don’t like using DNS to resolve names to addresses on routers, because of one nuisance feature on the router: Whenever you type a nonexistent

command on the router, the router assumes you are trying to telnet to a device by that *name* and tries to resolve it to an IP address. This is annoying because either you have to wait for the DNS query to time out or you must execute the break sequence (CTRL-SHIFT-6). You have another option, though, and that is to disable DNS lookups on the router with the following command:

```
Router(config)# no ip domain-lookup
```



5.27. The CD includes a multimedia demonstration of using and disabling name resolution on a Cisco router.

Verifying a Router's Operation

Once you have configured your router, there are many, many commands that you have available to use to examine and troubleshoot your configuration. This chapter covers some of the basic **show** commands that you have at your disposal. Chapter 6 covers some more commands, including **ping**, **telnet**, **show cdp**, and **debug**.

The **show interfaces** Command

One of the most common commands that you will use on a router is the **show interfaces** command. This command allows you to see the status and configuration of your interfaces, as well as some statistical information. Here is the syntax of this command:

```
Router> show interfaces [type [slot_#/]port_#]
```

If you don't specify a specific interface, the router displays all of its interfaces—those enabled as well as those disabled. Here is an example of the output of this command:

```
Router# show interfaces ethernet 0
Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.1234
                                     (bia 0000.0c00.1234)
  Internet address is 172.16.16.2, subnet mask is 255.255.255.252
  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 4000 bits/sec, 8 packets/sec
```

```

2240375 packets input, 887359872 bytes, 0 no buffer
Received 722137 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
10137586 packets output, 897215078 bytes, 0 underruns
4 output errors, 1037 collisions, 3 interface resets,
0 restarts

```

The first line of output shows the status of the interface. This status was described previously, in the section “Verifying Switch Operation.” The second line is the MAC address of the Ethernet interface. The third line has the IP address and subnet mask configured on the interface.

exam

Watch

Make sure you understand the output of the `show interfaces` command, since this is a powerful layer-2 troubleshooting tool.

The fourth line has the MTU Ethernet frame size as well as the routing protocol metrics. (These metrics are discussed in more depth in Chapters 9, 10, and 11, which discuss routing protocols.) Notice the “BW” parameter in this line. Referred to as the bandwidth of the link, this is used by some routing protocols, such as IGRP, OSPF, and EIGRP, when making routing

decisions. For Ethernet, this is always 10,000 Kbps. You can change this value with the **bandwidth** command, discussed previously, in the section “Configuring the Bandwidth Parameter.” Table 5-6 explains some of the elements that you may see with the **show interfaces** command. Please note that depending on the type of interface, the information displayed in the **show interfaces** command may be slightly different.



5.28. The CD includes a multimedia demonstration of using the `show interfaces` command on a Cisco router.

The `show ip interface` Command

Another common command that you will use on a router is the **show ip interface** command. This command enables you to see the IP configuration of your router’s interfaces:

```
Router> show ip interface [type [slot_#/]port_#]
```

Here is an abbreviated output of the **show ip interface** command:

```

Router# show ip interface
Ethernet1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command

```

TABLE 5-6

Explanation of the Elements in the **show interfaces** Command

Element	Description
Address	The MAC address of the interface; BIA (burnt-in address) is the MAC address burnt into the Ethernet controller—this can be overridden with the mac-address command
Last input/output	The last time a packet was received on or sent out of the interface—can be used to determine if the interface is operating or not
Last clearing	Indicates the last time the clear counters command was executed on the interface
Output queue	Indicates the number of packets waiting to be sent out the interface—the number after the “/” is the maximum size of the queue and then the number of packets dropped because the queue was full
Input queue	Indicates the number of packets received on the interface and waiting to be processed—the number after the “/” is the maximum size of the queue and then the number of packets dropped because the queue was full
No buffers (input)	Number of received packets dropped because the input buffer was filled up
Runts (input)	Number of packets received that were less than 64 bytes in length
Giants (input)	Number of packets received that were greater than the maximum allowed size—1518 bytes for Ethernet
Input errors	The total number of input errors received on the interface
CRC (input)	Indicates packets received that had checksum errors
Frame (input)	Indicates the number of packets received that had both CRC errors and cases where the length of the frame was not on a byte boundary.
Overruns (input)	Number of times the inbound packet rate exceeded the capabilities of the interface to process the traffic
Ignored (input)	Number of inbound packets that were dropped because of the lack of input buffer space
Aborts (input)	Number of received packets that were aborted
Collisions (output)	Number of times the interface tried transmitting a packet, but a collision occurred—this should be less than 0.1% of total traffic leaving the interface
Interface resets (output)	Number of times the interface changed state by going down and then coming back up
Restarts (output)	Number of times the controller was reset because of errors—use the show controllers command to troubleshoot this problem

```

MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access list is not set
Inbound access list is 100
<--output omitted-->

```

As you can see from this command, the IP address and mask are displayed, as well as the status of direct broadcast forwarding. Any access list applied to the interface is also displayed. Access lists are explained in Chapter 13.

exam

Watch

Use the `show ip interface` command to determine if an ACL is applied to an interface.

An additional parameter to the preceding command, **brief**, will display a single-line description for each interface, as shown here:

```

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 administratively down down

```

This is an extremely useful command when you want to see a quick overview of all of the interfaces on the router, their IP addresses, and their statuses.



5.29. The CD includes a multimedia demonstration of using the `show ip interface` command on a Cisco router.

The show hosts Command

To view the static and dynamic DNS entries in your router's resolution table, use this command:

```

Router# show hosts
Default domain is DOO.COM
Name/address lookup uses domain service
Name servers are 255.255.255.255
Host          Flag          Age  Type  Address(es)
a.check.com   (temp, OK)   1    IP    172.16.9.9
b.check.com   (temp, OK)   8    IP    172.16.1.1
f.check.com   (perm, OK)   0    IP    172.16.1.2

```

The first two entries in the table were learned via a DNS server (*temp* flag), whereas the last entry was configured statically on the router with the **ip host** command (*perm* flag).

The `show version` Command

If you want to see general information about your router—its model number, the types of interfaces, the different kinds and amounts of memory, its software version, where the router located and loaded its IOS and configuration, as well as the configuration settings—then you will want to use the **show version** command:

```
Router> show version
Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-JS-M), Version 12.0(3c), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Tue 13-Apr-99 07:39 by phanguye
Image text-base: 0x60008918, data-base: 0x60BDC000

ROM: System Bootstrap, Version 11.1(20)AA2, EARLY DEPLOYMENT
RELEASE SOFTWARE (fc1)

Router uptime is 2 days, 11 hours, 40 minutes
System restarted by power-on
System image file is "flash:c3640-js-mz.120-3c.bin"

cisco 3640 (R4700) processor (revision 0x00) with 49152K/16384K
bytes of memory.
<--output omitted-->
1 FastEthernet/IEEE 802.3 interface(s)
8 Low-speed serial(sync/async) network interface(s)
1 Channelized T1/PRI port(s)
DRAM configuration is 64 bits wide with parity disabled.
125K bytes of non-volatile configuration memory.
32768K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

At the very bottom of this output is the configuration register value. This value determines how the router will boot up, including how it will find its IOS and its configuration file. The bootup process of the router is discussed in much more depth in Chapter 6.



5.30. The CD includes a multimedia demonstration of using the `show version` command on a Cisco router.

exam**Watch**

Be familiar with the output of the `show version` command, including what is displayed, like the IOS version, the uptime, the amount of RAM, NVRAM, and flash, the type and number of interfaces, and the configuration register value.

Router Configuration Files

Working with configuration files on a router is exactly the same as working on a 2950 switch. To view your running (active) configuration file, use the **show running-config** command:

```
Router# show running-config
Building configuration...
Current configuration:
!
version 12.0
no service udp-small-servers
no service tcp-small-servers
!
hostname Router
<--output omitted-->
```

Notice the references to “Building configuration. . .” and “Current configuration” in this example. Both of these refer to the configuration in RAM.

To save your configuration file from RAM to NVRAM, use the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
Building configuration..
Router#
```

To see the configuration file stored in NVRAM, use this command:

```
Router# show startup-config
Using 4224 out of 65536 bytes
!
```

```

version 11.3
no service udp-small-servers
no service tcp-small-servers
!
hostname Router
<--output omitted-->

```

One difference between this output and that from the **show running-config** command is the first line of output: Using 4224 out of 65536 bytes. This refers to the amount of NVRAM currently used by the saved configuration file. As you can see from these commands, the syntax and operation are the same on the 2950.

exam

Watch

Unlike the 1900 switch, the router does not automatically save configuration changes. You must manually enter the *copy running-config startup-config* command from

Privileged EXEC mode. This command backs up your configuration to NVRAM. Also, anytime you want to examine or manipulate a configuration, you must be in Privilege EXEC mode.

EXERCISE 5-4



Using the CLI to Set Up a Basic Router Configuration

In this exercise, you will use the CLI to create a basic configuration on your router. In this exercise, you'll be creating a basic configuration on both the 2600 and 2500 routers in the simulator on the CD-ROM. You can find a picture of the network diagram for the simulator in the Introduction to this book. This exercise relies on Exercise 5-2 and 5-3 having been completed successfully (without shutting down the simulator). If you have not completed these Exercises, please do so now.

1. Starting with the 2600 router, go to *Privilege EXEC* mode and then enter *Configuration* mode.

At the top of the simulator's tool bar, click on the *eRouters* icon and select 2600. On the 2600 router, access *User EXEC* mode and use **enable** to go to *Privilege EXEC* mode and then **configure terminal** to access *Configuration* mode.

2. Assign a hostname of 2600.
Use the **hostname 2600** command and examine the prompt.
3. Set a *User EXEC* password of *cisco* for telnet access.
Enter the **line vty 0 4** command and follow it with the **login** and **password cisco** commands.
4. Assign an encrypted *Privilege EXEC* password of *cisco*.
Enter **enable secret cisco**.
5. Assign an IP address of 192.168.1.1/24 to **fastethernet 0/0**. Enable the interface. Test connectivity by pinging 2950-1.
Enter the interface with **interface fastethernet 0/0**. Next, enter the addressing information: **ip address 192.168.1.1 255.255.255.0**. Use the **no shutdown** command to enable the interface. Return to *Privilege EXEC* mode: **end**. Test connectivity: **ping 192.168.1.4**.
6. Check whether or not the **serial 0** interface is DTE or DCE. If it is DCE, assign a clock rate of 64,000. Assign an IP address of 192.168.2.1/24 to **serial 0**. Enable the interface.
Check the controller: **show controller serial 0**. Enter the interface with **configure terminal** and **interface serial 0**. Configure the clock rate: **clock rate 64000**. Next, enter the addressing information: **ip address 192.168.2.1 255.255.255.0**. Use the **no shutdown** command to enable the interface. Exit to *Privilege EXEC* mode: **end**.
7. Use a command to display all of the interfaces, their IP addresses, and their statuses on one screen and then show the details of *fastethernet0* and *serial0* separately.
Use the **show interfaces** command followed by **show interface fa0/0** and **show interface s0**.
8. Save your configuration to NVRAM and view the active configuration.
Save it with **copy running-config startup-config** and view it with **show running-config**.
9. Configure Host3's IP address (192.168.3.2/24) and default gateway (192.168.3.1). Configure Host2's IP address (192.168.3.3/24) and default gateway (192.168.3.1). Test connectivity between the two PCs.
Configure Host3: Click the *eStations* icon in the toolbar and select *Host3* within the simulator. On the CLI, type **ipconfig /ip 192.168.3.3**

255.255.255.0. Then, type **ipconfig /dg 192.168.3.1**. Configure Host2: Click the *eStations* icon within the simulator and select *Host2*. On the CLI, type **ipconfig /ip 192.168.3.2 255.255.255.0**. Then, type **ipconfig /dg 192.168.3.1**. Test the connection from Host2 by pinging Host3: **ping 192.168.3.2**.

10. Configure the 2500 router. Set the hostname (2500) as well as the telnet (*cisco*) and enable secret passwords (*cisco*); for the interfaces, enable *ethernet0* and *serial0* and assign the appropriate IP addresses to them. Test connectivity to the 2600 and Host-3 and Host-2.

At the top of the simulator's tool bar, click on the *eRouters* icon and select 2500. On the 2500 router, access *User EXEC* mode and use **enable** to go to *Privilege EXEC* mode and then **configure terminal** to access *Configuration* mode. Use the **hostname 2500** command and examine the prompt. Enter the **line vty 0 4** command and follow it with the **login** and **password cisco** commands. Enter **enable secret cisco**. Enter the interface with **interface ethernet 0**. Next, enter the addressing information: **ip address 192.168.3.1 255.255.255.0**. Use the **no shutdown** command to enable the interface. Exit *Configuration* mode (**end**) and check the *serial0* controller: **show controller serial 0**. Enter the interface with **configure terminal** and **interface serial 0**. It's a DTE, so you don't have to configure the clock rate. Next, enter the addressing information: **ip address 192.168.2.2 255.255.255.0**. Use the **no shutdown** command to enable the interface. Exit *Configuration* mode: **end**. Test connectivity to the 2600: **ping 192.168.2.1**. Test connectivity to the two hosts connected to *ethernet0*: **ping 192.168.3.2** and **ping 192.168.3.3**. Save the configuration: **copy running-config startup-config**.

You should now be more comfortable with creating a basic configuration on a router from the CLI. You do not need to save your configuration: actually, this requires the full-functioning version of Boson's NetSim™. However, the remaining chapters will build upon the configuration in this chapter. In subsequent chapters, you can either repeat the labs in Chapter 5 or have this basic configuration automatically loaded into the simulator.

CERTIFICATION SUMMARY

The IOS provides a CLI with many features, including context-sensitive help with **help** or **?**, command history, and advanced editing features, which are available at all modes, including the *User* and *Privilege EXEC* and *Configuration* modes. Within *Configuration* mode, there are *Subconfiguration* modes. To access *Privilege EXEC* mode, use the **enable** command, and to access *Configuration* mode, use **configure terminal**.

You can protect access to your Cisco device by assigning *User EXEC* and *Privilege EXEC* passwords. On the 1900, use the **enable password level** command. The router and 2950 use the *Line Subconfiguration* mode **password** command and the **enable secret** or **enable password** command for the two respective levels.

The 1900 has a global IP address, whereas the 2950 has its IP address associated with a VLAN interface. The router needs an IP address on each interface where it will be processing IP traffic. All of these devices use the **ip address** command. For DCE serial interfaces, routers need a clock rate applied with the **clock rate** command. Use the **show interfaces** command to view the status and configuration of your interfaces.

To view the active configuration, use the **show running-config** command. On the 2950 and router, you must manually save your configuration to NVRAM with the **copy running-config startup-config** command.



TWO-MINUTE DRILL

CLI

- There are three basic CLI modes for IOS devices: *User EXEC* (“>”), *Privilege EXEC* (“#”), and *Configuration* (“(config)#”) modes.
- On the 1900, you must enter **K** from the main menu to access *User EXEC* mode.
- Use the **enable** command to access *Privilege EXEC* mode and **configure terminal** to access *Configuration* mode. Use **end** to exit this mode and return to *Privilege EXEC* mode and either **exit** or **logout** to log out of the IOS device.
- Access context-sensitive help with the **help** or **?**.

Switch Basics

- The **hostname** command assigns a name to any IOS device.
- Use the **enable password level** command to assign passwords to the 1900.
- The **shutdown** command disables an interface and **no shutdown** enables it; switch interfaces are enabled by default and router interfaces are disabled by default.
- On the 2950, you must be in a VLAN interface in order to assign an IP address.
- Use the **ip default-gateway** command to assign an exit point to the subnet.
- Use the **show running-config** to view your switch’s configuration.
- The 1900 automatically saves its configuration to NVRAM. The 2950 requires the **copy running-config startup-config** command in order to save its configuration.

Router Basics

- The *System Configuration Dialog* is presented automatically if a router boots up without a configuration file in NVRAM or if you execute the **setup** *Privilege EXEC* mode command.

- ❑ You can assign *User EXEC* passwords to restrict line access (console and VTY) with the **password** command. Use the **enable secret** command to encrypt the *Privilege EXEC* password. The **enable secret** command doesn't encrypt it.
- ❑ DCE interfaces require a clock rate, assigned with the **clock rate** command. Use the **show controller** command to verify the interface type. The **bandwidth** command is used only for routing metrics, not clocking.
- ❑ Use the **show interfaces** command to see the physical and data link layer statuses.

SELF TEST

The following Self Test questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question.

IOS Introduction

1. Which command takes you from *Privilege EXEC* mode to *User EXEC* mode?
 - A. enable
 - B. disable
 - C. exit
 - D. logout
2. Which prompt indicates that you are at *User EXEC* mode?
 - A. >
 - B. #
 - C. %
 - D. @
3. The _____ command takes you from *User EXEC* mode to *Privilege EXEC* mode.

IOS Basics

4. If you wanted a list of parameters for the **show** command, you would type _____.
5. To recall a previous command, which of the following would you use?
 - A. CTRL-P
 - B. ctrl-n
 - C. ctrl-r
 - D. down arrow
6. Which control sequence moves the cursor to the end of the line?
 - A. CTRL-A
 - B. ctrl-p
 - C. ctrl-e
 - D. ctrl-z

7. What would you enter to see the last few commands you entered?
 - A. `show version`
 - B. `show commands`
 - C. `show previous`
 - D. `show history`
8. Which would you use to move your cursor back one word?
 - A. ESC-B
 - B. `esc-p`
 - C. `ctrl-b`
 - D. `ctrl-p`
9. If you enter a command in either *User EXEC* mode or *Privilege EXEC* mode that the IOS does not recognize, the switch or router
 - A. Logs you out
 - B. Tries to resolve the command to an IP address
 - C. Executes the closest command that it can find in its command set
 - D. Makes a “beep” sound
10. You are at *User EXEC* mode and type the letter *e*. What message appears?
 - A. `% Incomplete command.`
 - B. `% Ambiguous command:`
 - C. `% Unknown command or computer name`
 - D. `% Invalid input detected at '^'.`

Basic Switch Configuration

11. On the 1900 switch, how do you enter *User EXEC* mode?
 - A. It comes up by default.
 - B. Enter the letter **U** in the menu.
 - C. Enter the letter **K** in the menu.
 - D. Enter the letter **C** in the menu.
12. Enter the command to access *Configuration* mode: _____.
13. Enter the command to assign a name to a 1900 switch of *switch1*: _____.

14. What command assigns a *User EXEC* password to a 2950 switch?
- A. `password`
 - B. `login`
 - C. `enable password`
 - D. `enable password level`
15. Which command assigns a default gateway to the 2950 switch?
- A. `(config)# default-gateway`
 - B. `(config-if)# default-gateway`
 - C. `(config-if)# ip default-gateway`
 - D. `(config)# ip default-gateway`

Basic Router Configuration

16. The router automatically saves its configuration.
- A. True
 - B. False
17. You examine your interfaces, and the Ethernet 0 interface status says: `Ethernet 0 is up, line protocol is down`. What does this indicate?
- A. There is a physical layer problem.
 - B. There is a data link layer problem.
 - C. There is a network layer problem.
 - D. There is no problem.
18. You execute the **line** `console 0` command from *Configuration* mode. What will the router's prompt be?
- A. `Router(config)#`
 - B. `Router(config-line)#`
 - C. `Router(config-interface)#`
 - D. `Router#(config-if)`
19. You are in *Interface Subconfiguration* mode and want to completely return to *Privilege EXEC* mode. What would you enter? (Choose all correct answers.)
- A. CTRL-Z
 - B. `end`
 - C. `exit`
 - D. `logout`

- 20.** What router command saves the active configuration to NVRAM?
- A. `copy nvram startup-config`
 - B. `copy startup-config running-config`
 - C. `copy running-config nvram`
 - D. `copy running-config startup-config`

SELF TEST ANSWERS

IOS Introduction

- B.** The **disable** command takes you from *Privilege EXEC* mode to *User EXEC* mode.
 A does the reverse and therefore is incorrect, and **C** and **D** log you out of the router.
- A.** The “>” indicates *User EXEC* mode.
 B is incorrect because it indicates *Privilege EXEC* mode. **C** and **D** are incorrect because these are invalid prompts.
- The **enable** command takes you from *User EXEC* mode to *Privilege EXEC* mode.

IOS Basics

- To view all of the parameters for the **show** command, enter **show ?**.
- A.** CTRL-P recalls a previous command.
 B and **D** recall the next command, and **C** redisplay the current line.
- C.** CTRL-E takes you to the end of the CLI.
 A takes you to the beginning, **B** takes you to the previous line, and **E** logs you out.
- D.** The **show history** command displays the last few commands that you entered.
 A displays software and hardware information. **B** and **D** are nonexistent commands.
- A.** The ESC-B control sequence takes you back one word on the CLI.
 B is incorrect because it is an invalid control sequence. **C** is incorrect because it takes you back one character. **D** is also incorrect, because it takes you back one line.
- B.** Whenever you enter an invalid command, the device attempts to first resolve the command to an IP address and telnet to this machine.
 A is incorrect because this requires you to execute the **logout** command. **C** is incorrect because the IOS doesn't know what command to execute. **D** is incorrect because sounds aren't supported with the CLI.
- B.** If you don't type enough characters for a command or parameter, you'll get an “ambiguous command” message.
 A is incorrect because you didn't type enough parameters for the command. **C** is incorrect because this message is shown when you enter an invalid command, and **D** is incorrect because you entered incorrect or invalid information for a valid command.

Basic Switch Configuration

11. C. Enter **K** to access *User EXEC* mode on a 1900 switch.
 A is true for the 2950 and the routers, not the 1900. **B** and **D** are incorrect because these are nonexistent options.
12. The **configure terminal** command takes you from *Privilege EXEC* mode to *Configuration* mode.
13. The **hostname** switch1 command changes the name of the 1900 switch to “switch1.”
14. A. The *Line Subconfiguration* mode **password** command assigns a console or telnet *User EXEC* mode password.
 B is incorrect because this specifies authentication with the **password** command. **C** is incorrect because this assigns a *Privilege EXEC* password. **D** is incorrect because this assigns passwords on a 1900 switch.
15. D. The **ip default-gateway** *Global Configuration* mode command assigns a default gateway to either a 1900 or 2950 switch.
 C is incorrect because it is in the wrong mode. **B** and **D** are incorrect because they are invalid commands.

Basic Router Configuration

16. B. The router does not automatically save its configuration; you need to use the **copy running-config startup-config** to save it to NVRAM.
17. B. The “line protocol is down” refers to the data link layer.
 A is incorrect because the physical layer is “up.” **C** is incorrect because the status refers only to the physical and data link layers. Since there is a correct answer, **D** is incorrect.
18. B. When you execute the **line console 0** command, your prompt changes to “Router(config-line)#.”
 A is incorrect because this is *Global Configuration* mode. **D** represents *Interface Subconfiguration* mode, and **C** is a nonexistent prompt.
19. A and B. The CTRL-Z control sequence and the **end** command take you from any *Configuration* mode back to *Privilege EXEC* mode.
 C only takes you back one level, while **D** is a *User* and *Privilege EXEC* mode command.
20. D. Use the **copy running-config startup-config** to back up your configuration from RAM to NVRAM.
 B is incorrect because this restores your configuration to RAM. **A** and **C** are incorrect because these commands use nonexistent parameters.