Administering the WMIC

This chapter describes how to administer the Cisco Wireless Mobile Interface (WMIC).

Configuring a System Name and Prompt

You configure the system name on the WMIC to identify it. A “greater than” symbol (>) is appended. The prompt is updated whenever the system name changes, unless you manually configure the prompt by using the **prompt** command in global configuration mode.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference* and the *Cisco IOS IP and IP Routing Command Reference for Release 12.1*.

Configuring a System Name

To manually configure a system name, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**hostname** *name*	Manually configure a system name. The name must follow the rules for ARPANET hostnames. They must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphens. Names can be up to 63 characters.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

When you set the system name, it is also used as the system prompt.

To return to the default hostname, use the **no hostname** global configuration command.

Managing DNS

The DNS protocol controls the Domain Name System (DNS), a distributed database with which you can map hostnames to IP addresses. When you configure DNS on your WMIC, you can substitute the hostname for the IP address with all IP commands, such as **ping**, **telnet**, **connect**, and related Telnet support operations.

IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco Systems is a commercial organization that is identified by a *com* domain name; its domain name is *cisco.com*.

To keep track of domain names, IP has defined the concept of a domain name server that holds a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, identify the hostnames, specify the name server that is present on your network, and enable the DNS.

Default DNS Configuration

Table 4-1 shows the default DNS configuration.

Table 4-1 Default DNS Configuration

Feature	Default Setting
DNS enable state	Disabled.
DNS default domain name	None configured.
DNS servers	No name server addresses are configured.

Setting Up DNS

To set up your WMIC to use the DNS, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip domain-name** *name*	Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name). Do not include the initial period that separates an unqualified name from the domain name. At boot time, no domain name is configured; however, if the configuration comes from a BOOTP or Dynamic Host Configuration Protocol (DHCP) server, then the default domain name might be set by the BOOTP or DHCP server (if the servers were configured with this information).
Step 3	**ip name-server** *server-address1* [*server-address2 ... server-address6*]	Specifies the address of one or more name servers to use for name and address resolution. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The WMIC sends DNS queries to the primary server first. If that query fails, the backup servers are queried.

	Command	Purpose
Step 4	**ip domain-lookup**	(Optional) Enables DNS-based hostname-to-address translation on your WMIC. This feature is enabled by default. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS).
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

If you use the WMIC IP address as its hostname, the IP address is used and no DNS query occurs. If you configure a hostname that contains no periods (.), a period followed by the default domain name is appended to the hostname before the DNS query is made to map the name to an IP address.

The default domain name is the value set by the **ip domain-name** global configuration command. If there is a period (.) in the hostname, the Cisco IOS software looks up the IP address without appending any default domain name to the hostname.

To remove a domain name, use the **no ip domain-name** *name* command in global configuration mode. To remove a name server address, use the **no ip name-server** *server-address* command in global configuration mode. To disable DNS on the WMIC, use the **no ip domain-lookup** command in global configuration mode.

Displaying the DNS Configuration

To display the DNS configuration information, use the **show running-config** command in privileged EXEC command.

Creating a Banner

You can configure a message-of-the-day (MOTD) and a login banner. The MOTD banner appears on all connected terminals at login and is useful for sending messages that affect all network users (such as impending system shutdowns).

The login banner also appears on all connected terminals. It appears after the MOTD banner and before the login prompts.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.2*.

Default Banner Configuration

The MOTD and login banners are not configured.

Configuring a Message-of-the-Day Login Banner

You can create a single- or multiple-line message banner that appears on the screen when someone logs in to the WMIC.

To configure a MOTD login banner, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**banner motd** *c message c*	Specifies the message of the day. For *c*, enter the delimiting character of your choice, such as a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded. For *message*, enter a banner message up to 255 characters. You cannot use the delimiting character in the message.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete the MOTD banner, use the **no banner motd** global configuration command.

This example shows how to configure a MOTD banner for the WMIC by using the pound sign (#) as the beginning and ending delimiter:

```
bridge(config)# banner motd #
This is a secure site. Only authorized users are allowed.
For access, contact technical support.
#
bridge(config)#
```

This example shows the banner displayed from the previous configuration:

```
Unix> telnet 172.2.5.4
Trying 172.2.5.4...
Connected to 172.2.5.4.
Escape character is '^]'.

This is a secure site. Only authorized users are allowed.
For access, contact technical support.

User Access Verification

Password:
```

Configuring a Login Banner

You can configure a login banner to appear on all connected terminals. This banner appears after the MOTD banner and before the login prompt.

To configure a login banner, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**banner login** *c message c*	Specifies the login message. For *c*, enter the delimiting character of your choice, such as a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded. For *message*, enter a login message up to 255 characters. You cannot use the delimiting character in the message.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete the login banner, use the **no banner login** global configuration command.

This example shows how to configure a login banner for the WMIC using the dollar sign ($) symbol as the beginning and ending delimiter:

```
bridge(config)# banner login $
Access for authorized users only. Please enter your username and password.
$
bridge(config)#
```

Protecting Access to Privileged EXEC Commands

A simple way of controlling terminal access in your network is to use passwords and assign privilege levels. Password protection restricts access to a network or network device. Privilege levels define what commands users can issue after they have logged into a network device.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference for Release 12.2*.

This section describes how to control access to the configuration file and privileged EXEC commands.

Default Password and Privilege Level Configuration

Table 4-2 shows the default password and privilege level configuration.

Table 4-2 Default Password and Privilege Levels

Feature	Default Setting
Username and password	Default username is *Cisco* and the default password is *Cisco.*
Enable password and privilege level	Default password is *Cisco*. The default is level 15 (privileged EXEC level). The password is encrypted in the configuration file.
Enable secret password and privilege level	The default enable password is *Cisco*. The default is level 15 (privileged EXEC level). The password is encrypted before it is written to the configuration file.
Line password	Default password is *Cisco*. The password is encrypted in the configuration file.

Setting or Changing a Static Enable Password

The enable password controls access to the privileged EXEC mode.

Note The **no enable password** global configuration command removes the enable password, but you should use extreme care when using this command. If you remove the enable password, you are locked out of the EXEC mode.

To set or change a static enable password, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**enable password** *password*	Defines a new password or change an existing password for access to privileged EXEC mode. The default password is *Cisco*. For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, it is case sensitive, and it allows spaces but ignores leading spaces. It can contain the question mark (?) character if you precede the question mark with the key combination Crtl-V when you create the password; for example, to create the password abc?123, do this: 1. Enter **abc**. 2. Enter **Crtl-V**. 3. Enter **?123**. When the system prompts you to enter the enable password, you need not precede the question mark with the Ctrl-V; you can simply enter abc?123 at the password prompt.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file. The enable password is not encrypted and can be read in the WMIC configuration file.

This example shows how to change the enable password to *l1u2c3k4y5*. The password is not encrypted and provides access to level 15 (traditional privileged EXEC mode access).

```
bridge(config)# enable password l1u2c3k4y5
```

Protecting Enable and Enable Secret Passwords with Encryption

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a Trivial File Transfer Protocol (TFTP) server, you can use either the **enable password** or the **enable secret** command. Both commands accomplish the same thing; that is, you can establish an encrypted password that users must enter to access privileged EXEC mode (the default) or any privilege level you specify.

We recommend that you use the **enable secret** command because it uses an improved encryption algorithm.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure encryption for enable and enable secret passwords, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**enable password** [**level** *level*] {*password* | *encryption-type encrypted-password*}	Defines a new password or change an existing password for access to privileged EXEC mode.
	or	or
	enable secret [**level** *level*] {*password* | *encryption-type encrypted-password*}	Defines a secret password, which is saved using a nonreversible encryption method. • (Optional) For *level*, the range is from 0 to 15. Level 1 is normal user EXEC mode privileges. The default level is 15 (privileged EXEC mode privileges). • For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, it is case sensitive, and it allows spaces but ignores leading spaces. By default, no password is defined. • (Optional) For *encryption-type*, only type 5, a Cisco proprietary encryption algorithm, is available. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another WMIC configuration. **Note** If you specify an encryption type and then enter a clear text password, you can not reenter privileged EXEC mode. You cannot recover a lost encrypted password by any method.
Step 3	**service password-encryption**	(Optional) Encrypt the password when the password is defined or when the configuration is written. Encryption prevents the password from being readable in the configuration file.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

If both the enable and enable secret passwords are defined, users must enter the enable secret password.

Use the **level** keyword to define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. To specify commands accessible at various levels, use the **privilege level** command in global configuration mode. For more information, see the "Configuring Multiple Privilege Levels" section on page 4-10.

If you enable password encryption, it applies to all passwords, including username passwords, authentication key passwords, the privileged command password, and console and virtual terminal line passwords.

To remove a password and level, use the **no enable password** [**level** *level*] or **no enable secret** [**level** *level*] command in global configuration mode. To disable password encryption, use the **no service password-encryption** command in global configuration mode.

This example shows how to configure the encrypted password *1FaD0$Xyti5Rkls3LoyxzS8* for privilege level 2:

```
bridge(config)# enable secret level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8
```

Configuring Username and Password Pairs

You can configure username and password pairs, which are locally stored on the WMIC. These pairs are assigned to lines or interfaces, and they authenticate each user before that user can access the WMIC. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

To establish a username-based authentication system that requests a login username and a password, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}	Enters the username, privilege level, and password for each user. • For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed. • (Optional) For *level*, specify the privilege level the user has after gaining access. The range is from 0 to 15. Level 15 gives privileged EXEC mode access. Level 1 gives user EXEC mode access. • For *encryption-type*, enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow. • For *password*, specify the password the user must enter to gain access to the WMIC. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
Step 3	**login local**	Enables local password checking at login time. Authentication is based on the username specified in Step 2.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable username authentication for a specific user, use the **no username** *name* command in global configuration mode.

To disable password checking and allow connections without a password, use the **no login** command in line configuration mode.

Note You must have at least one username configured and you must set your local login to open a Telnet session to the WMIC. If you enter no username for the only username, you can be locked out of the WMIC.

Configuring Multiple Privilege Levels

By default, the Cisco IOS software has two modes of password security: user EXEC and privileged EXEC. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password fairly widely. But if you want fewer users to have access to the **configure** command, you can assign it level 3 security and distribute that password to a smaller group of users.

Setting the Privilege Level for a Command

To set the privilege level for a command mode, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**privilege** *mode* **level** *level command*	Sets the privilege level for a command. • For *mode*, enter **configure** for global configuration mode, **exec** for EXEC mode, **interface** for interface configuration mode, or **line** for line configuration mode. • For *level*, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. Level 15 is the level of access permitted by the **enable** password. • For *command*, specify the command to which you want to restrict access.
Step 3	**enable password level** *level password*	Specifies the enable password for the privilege level. • For *level*, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. • For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, it is case sensitive, and it allows spaces but ignores leading spaces. By default, no password is defined.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config** or **show privilege**	Verifies your entries. The **show running-config** command displays the password and access level configuration. The **show privilege** command displays the privilege level configuration.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

When you set a command to a privilege level, all commands whose syntax is a subset of that command are also set to that level. For example, if you set the **show ip route** command to level 15, the **show** commands and **show ip** commands are automatically set to privilege level 15 unless you set them individually to different levels.

To return to the default privilege for a given command, use the **no privilege** *mode* **level** *level command* command in global configuration mode.

This example shows how to set the **configure** command to privilege level 14 and define *SecretPswd14* as the password that users must enter to use level 14 commands:

```
bridge(config)# privilege exec level 14 configure
bridge(config)# enable password level 14 SecretPswd14
```

Logging Into and Exiting a Privilege Level

To log in to a specified privilege level and to exit to a specified privilege level, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**enable** *level*	Logs in to a specified privilege level. For *level*, the range is from 0 to 15.
Step 2	**disable** *level*	Exits to a specified privilege level. For *level*, the range is from 0 to 15.

Protecting the Wireless LAN

Configure security settings to prevent unauthorized access to your network. Because it is a radio device, the WMIC can communicate beyond the physical boundaries of your building. You can apply advanced security features such as the following:

- Unique service set identifiers (SSIDs) that are not broadcast in the beacon (see Chapter 5, “Configuring SSIDs”)
- Wired Equivalent Privacy (WEP) and WEP features (see Chapter 8, “Configuring Cipher Suites and WEP”)
- Dynamic WEP authentication (see Chapter 9, “Configuring Authentication Types”)

Using VLANs

Assign SSIDs to the VLANs on the wireless LAN. If you do not use VLANs on the wireless LAN, the security options that can be assigned to SSIDs are limited, because encryption settings and authentication types are linked. Without VLANs, encryption settings (WEP and ciphers) are applied to an interface, and no more than one encryption setting can be used on each interface.

For example, if an SSID with static WEP is created with VLANs disabled, an additional SSID with Wi-Fi Protected Access (WPA) authentication cannot be created because of the different encryption settings. If a security setting for an SSID conflicts with another SSID, delete one or more SSIDs to eliminate the conflict.

Express Security Types

Table 4-3 describes the four security types that you can assign to an SSID.

Table 4-3 Security Types Assignable to SSIDs

Security Type	Description	Security Features Enabled
No Security	This is the least secure option. Use this option only for SSIDs that are used in a public space. Assign this option to a VLAN that restricts access to your network.	None.
Static WEP Key	This option is more secure than no security. However, static WEP keys are vulnerable to attack. If you configure this settings, you should limit association to the access point based on MAC address, or, if the network does not have a RADIUS server, consider using an access point as a local authentication server.	Mandatory WEP encryption, no key management, and open authentication. In root access point mode, client devices cannot associate using this SSID without a WEP key that matches the access point key.

Table 4-3 Security Types Assignable to SSIDs (continued)

Security Type	Description	Security Features Enabled
Extensible Authentication Protocol (EAP) Authentication	This option enables 802.1x authentication (such as LEAP, PEAP, EAP-TLS, EAP-TTLS, EAP-GTC, and other 802.1X/EAP based products). It requires an IP address and shared secret for an authentication server on the network (server authentication port 1645). Because 802.1x authentication provides dynamic encryption keys, a WEP key is not required.	Mandatory 802.1x authentication, In root access point mode, client devices that associate using this SSID must perform 802.1x authentication. If radio clients are configured to authenticate using EAP-FAST, open authentication with EAP should also be configured. If you do not configure open authentication with EAP, the following GUI warning message appears: WARNING: Network EAP is used for LEAP authentication only. If radio clients are configured to authenticate using EAP-FAST, Open Authentication with EAP should also be configured. If you are using the command-line interface (CLI), this warning message appears: SSID CONFIG WARNING: [SSID]: If radio clients are using EAP-FAST, AUTH OPEN with EAP should also be configured.
WPA	WPA permits wireless access to users authenticated against a database through the services of an authentication server, and encrypts those users' IP traffic with stronger algorithms than those used in WEP. As with EAP authentication, the IP address and shared secret for an authentication server on your network (server authentication port 1645) are required. This setting uses encryption ciphers, Temporal Key Integrity Protocol (TKIP), open authentication + EAP, network EAP authentication, key management WPA mandatory, and RADIUS server authentication port 1645.	Mandatory WPA authentication. Client devices that associate using this SSID must be WPA-capable. If radio clients are configured to authenticate using EAP-FAST, open authentication with EAP should also be configured. If you don't configure open authentication with EAP, the following GUI warning message appears: WARNING: Network EAP is used for LEAP authentication only. If radio clients are configured to authenticate using EAP-FAST, Open Authentication with EAP should also be configured. If you are using the CLI, this warning message appears: SSID CONFIG WARNING: [SSID]: If radio clients are using EAP-FAST, AUTH OPEN with EAP should also be configured.

Security Configuration Examples

This section contains these configuration examples:

- No Security SSID Example
- Static WEP Security Example
- EAP Authentication Security Example
- WPA Security Example

No Security SSID Example

This example shows part of the configuration for creating an SSID called *no_security_ssid*, including the SSID in the beacon, assigning it to VLAN 10, and selecting VLAN 10 as the native VLAN (as it applies to the 2.4-GHz ([802.11b/g]) WMIC):

```
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 ssid no_security-ssid
    vlan 10
    authentication open
    guest-mode
!
 concatenation
 speed basic-1.0 basic-2.0 basic-5.5 6.0 9.0 basic-11.0 12.0 18.0 24.0 36.0 48.0 54.0
 rts threshold 4000
 station-role root
 infrastructure-client
 bridge-group 1
!
interface Dot11Radio0.10
 encapsulation dot1Q 10
 no ip route-cache
 bridge-group 10
 bridge-group 10 spanning-disabled
!
interface FastEthernet0
 no ip address
 no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
!
interface FastEthernet0.10
      encapsulation dot1Q 10
      no ip address
      no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
```

As it applies to the 4.9-GHz WMIC:

```
hostname root
!
username Cisco password 7 02250D480809
ip subnet-zero
!
no aaa new-model
!
```

```
bridge irb
!
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 ssid test
    authentication open
    infrastructure-ssid
 !
 spacing 5 channel 4942
 speed basic-1.5 2.25 basic-3.0 4.5 basic-6.0 9.0 12.0 13.5
 power local 10
 station-role root
 infrastructure-client
 bridge-group 1
 bridge-group 1 spanning-disabled
!
interface FastEthernet0
 no ip address
 no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
 bridge-group 1 spanning-disabled
!
interface BVI1
 ip address 192.1.1.2 255.255.255.0
 no ip route-cache
!
ip http server
no ip http secure-server
ip http help-path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag
ip radius source-interface BVI1
logging snmp-trap emergencies
logging snmp-trap alerts
logging snmp-trap critical
logging snmp-trap errors
logging snmp-trap warnings
bridge 1 route ip
!
!
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line vty 0 4
 login local
 transport preferred all
 transport input all
 transport output all
line vty 5 15
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Static WEP Security Example

This example shows part of the configuration for creating an SSID called *static_wep_ssid*, excluding the SSID from the beacon, assigning the SSID to VLAN 20, selecting 3 as the key slot, and entering a 128-bit key:

```
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 encryption vlan 20 key 3 size 128bit 7 4E78330C1A841439656A9323F25A transmit-key
 encryption vlan 20 mode wep mandatory
 !
 ssid static_wep_ssid
    vlan 20
    authentication open
 !
 concatenation
 speed basic-1.0 basic-2.0 basic-5.5 6.0 9.0 basic-11.0 12.0 18.0 24.0 36.0 48.0 54.0
 rts threshold 4000
 station-role root
 infrastructure-client
 bridge-group 1
!
interface Dot11Radio0.20
 encapsulation dot1Q 20
 no ip route-cache
 bridge-group 20
 bridge-group 20 spanning-disabled
!
interface FastEthernet0
 no ip address
 no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
!
interface FastEthernet0.20
 encapsulation dot1Q 20
 no ip route-cache
 bridge-group 20
 bridge-group 20 spanning-disabled
```

EAP Authentication Security Example

This example shows part of the configuration for creating an SSID called *eap_ssid*, excluding the SSID from the beacon, and assigning the SSID to VLAN 30:

```
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 encryption vlan 30 mode wep mandatory
 !
 ssid eap_ssid
    vlan 30
    authentication open eap eap_methods
    authentication network-eap eap_methods
 !
 speed basic-1.0 basic-2.0 basic-5.5 basic-11.0
 rts threshold 2312
 station-role root
 bridge-group 1
```

```
 bridge-group 1 subscriber-loop-control
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding
 bridge-group 1 spanning-disabled
!
interface Dot11Radio0.30
 encapsulation dot1Q 30
 no ip route-cache
 bridge-group 30
 bridge-group 30 subscriber-loop-control
 bridge-group 30 block-unknown-source
 no bridge-group 30 source-learning
 no bridge-group 30 unicast-flooding
 bridge-group 30 spanning-disabled
!
interface FastEthernet0
 mtu 1500
 no ip address
 ip mtu 1564
 no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
 no bridge-group 1 source-learning
 bridge-group 1 spanning-disabled
!
interface FastEthernet0.30
 mtu 1500
 encapsulation dot1Q 30
 no ip route-cache
 bridge-group 30
 no bridge-group 30 source-learning
 bridge-group 30 spanning-disabled
!
```

WPA Security Example

This example shows part of the configuration for creating an SSID called *wpa_ssid*, excluding the SSID from the beacon, and assigning the SSID to VLAN 40:

```
aaa new-model
!
aaa group server radius rad_eap
 server 10.91.104.92 auth-port 1645 acct-port 1646
!
aaa group server radius rad_mac
!
aaa group server radius rad_acct
!
aaa group server radius rad_admin
!
aaa group server tacacs+ tac_admin
!
aaa group server radius rad_pmip
!
aaa group server radius dummy
!
aaa authentication login eap_methods group rad_eap
aaa authentication login mac_methods local
aaa authorization exec default local
aaa authorization ipmobile default group rad_pmip
aaa accounting network acct_methods start-stop group rad_acct
```

```
aaa session-id common
!
!
bridge irb
!
!
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 encryption vlan 40 mode ciphers tkip
!
 ssid wpa_ssid
    vlan 40
    authentication open eap eap_methods
    authentication network-eap eap_methods
    authentication key-management wpa
!
 concatenation
 speed basic-1.0 basic-2.0 basic-5.5 6.0 9.0 basic-11.0 12.0 18.0 24.0 36.0 48 54.0
 rts threshold 4000
 station-role root
 infrastructure-client
 bridge-group 1
!
interface Dot11Radio0.40
 encapsulation dot1Q 40
 no ip route-cache
 bridge-group 40
!
interface FastEthernet0
 no ip address
 no ip route-cache
 duplex auto
 speed auto
 bridge-group 1
!
interface FastEthernet0.40
 encapsulation dot1Q 40
 no ip route-cache
 bridge-group 40
!
ip http server
ip http help-path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag
/122-15.JA/1100
ip radius source-interface BVI1
radius-server attribute 32 include-in-access-req format%h
radius-server host 10.91.104.92 auth-port 1645 acct-port 1646 key 7 135445415F59
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
bridge 1 route ip
!
line con 0
line vty 5 15
!
end
```

Configuring and Enabling RADIUS

This section describes how to configure and enable Remote Authentication Dial-In User Service (RADIUS).

Understanding RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software from Cisco, Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.

Use RADIUS in these network environments:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that is customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as an access environment that uses a *smart card* access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and to grant access to network resources.
- Networks already using RADIUS. You can add a Cisco bridge containing a RADIUS client to the network.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, showing the amount of resources (such as time, packets, bytes, and so forth) used during the session. An Internet service provider might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

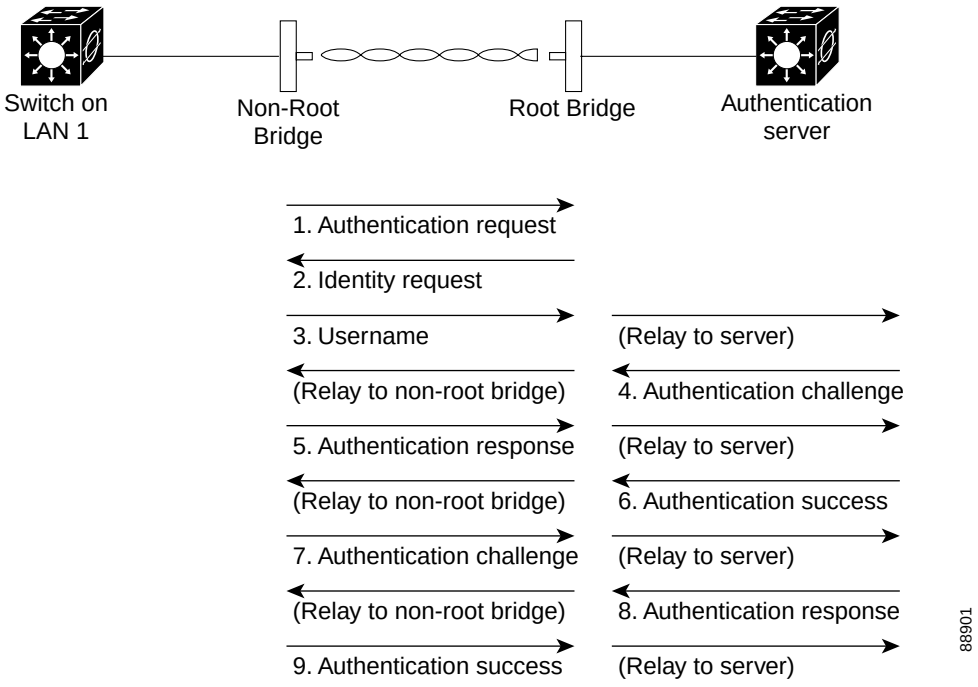
RADIUS is not suitable for these network situations:

- Multiprotocol access environments. RADIUS does not support AppleTalk Remote Access (ARA), NetBIOS Frame Control Protocol (NBFCP), NetWare Asynchronous Services Interface (NASI), or X.25 Packet Assembler Disassembler (PAD) connections.
- Switch-to-switch or router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a non-root bridge attempts to authenticate to a bridge whose access is controlled by a RADIUS server, authentication to the network occurs in the steps shown in Figure 4-1.

Figure 4-1 Sequence for EAP Authentication



In Figure 4-1, a non-root bridge and a RADIUS server on the wired LAN use 802.1x and EAP to perform a mutual authentication through the root device. The RADIUS server sends an authentication challenge to the non-root bridge. The non-root bridge uses a one-way encryption of the user-supplied password to generate a response to the challenge and sends that response to the RADIUS server. Using information from its user database, the RADIUS server creates its own response and compares that to the response from the non-root bridge. When the RADIUS server authenticates the non-root bridge, the process repeats in reverse, and the non-root bridge authenticates the RADIUS server.

When mutual authentication is complete, the RADIUS server and the non-root bridge determine a WEP key that is unique to the non-root bridge and that provides the non-root bridge with the appropriate level of network access, thereby approximating the level of security in a wired switched segment to an individual desktop. The non-root bridge loads this key and prepares to use it for the logon session.

During the logon session, the RADIUS server encrypts and sends the WEP key, called a *session key*, over the wired LAN to the root device. The root device encrypts its broadcast key with the session key and sends the encrypted broadcast key to the non-root bridge, which uses the session key to decrypt it. The non-root bridge and the root device activate WEP and use the session and broadcast WEP keys for all communications during the remainder of the session.

There is more than one type of EAP authentication, but the root device behaves the same way for each type: it relays authentication messages from the non-root bridge to the RADIUS server and from the RADIUS server to the non-root bridge. See the "Assigning Authentication Types to an SSID" section on page 9-15 for instructions on setting up authentication using a RADIUS server.

Controlling WMIC Access with RADIUS

This section describes how to control administrator access to the WMIC using RADIUS.

RADIUS provides detailed accounting information and flexible administrative control over authentication and authorization processes. RADIUS is facilitated through AAA and can be enabled only through authentication, authorization, and accounting (AAA) commands. RADIUS and AAA are disabled by default.

At a minimum, the host or hosts that run the RADIUS server software must be identified and the method lists for RADIUS authentication must be defined. Optionally, method lists for RADIUS authorization and accounting can be defined.

A method list defines the sequence and methods to be used to authenticate, to authorize, or to keep accounts on a non-root bridge. Method lists are used to designate one or more security protocols to be used, thus ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on non-root bridges; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

You must have access to and should configure a RADIUS server before you configure RADIUS features.

These sections describe RADIUS configuration:

- Identifying the RADIUS Server Host
- Configuring RADIUS Login Authentication
- Defining AAA Server Groups
- Configuring RADIUS Authorization for User Privileged Access and Network Services
- Starting RADIUS Accounting
- Configuring Settings for All RADIUS Servers
- Configuring the Bridge to Use Vendor-Specific RADIUS Attributes
- Configuring the Bridge for Vendor-Proprietary RADIUS Server Communication
- Displaying the RADIUS Configuration

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference for Release 12.2*.

Identifying the RADIUS Server Host

Access point-to-RADIUS-server communication involves several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Key string
- Timeout period
- Retransmission value

RADIUS security servers are identified by their hostname or IP address, hostname and specific User Datagram Protocol (UDP) port numbers, or IP address and specific UDP port numbers. The combination of the IP address and the UDP port number creates a unique identifier allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address.

If two different host entries on the same RADIUS server are configured for the same service—such as accounting—the second host entry configured acts as a failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the bridge tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order that they are configured.)

A RADIUS server and the bridge use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host that is running the RADIUS server daemon and a secret text (key) string that it shares with the bridge.

The timeout, retransmission, and encryption key values can be configured globally per server for all RADIUS servers or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the bridge, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command in global configuration mode.

Note If you configure both global and per-server functions (timeout, retransmission, and key commands) on the bridge, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands. For information on configuring these setting on all RADIUS servers, see the "Configuring Settings for All RADIUS Servers" section on page 4-28.

You can configure the bridge to use AAA server groups to group existing server hosts for authentication. For more information, see the "Defining AAA Server Groups" section on page 4-25.

To configure per-server RADIUS server communication, follow these required steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.

	Command	Purpose
Step 3	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]	Specifies the IP address or hostname of the remote RADIUS server host. • (Optional) For **auth-port** *port-number*, specify the UDP destination port for authentication requests. • (Optional) For **acct-port** *port-number*, specify the UDP destination port for accounting requests. • (Optional) For **timeout** *seconds*, specify the time interval that the bridge waits for the RADIUS server to reply before retransmitting. The range is 1 to 1000. This setting overrides the interval set using the **radius-server timeout** command. If no timeout is set with the **radius-server host** command, the time interval set with the **radius-server timeout** command is used. • (Optional) For **retransmit** *retries*, specify the number of times a RADIUS request is resent to a server if that server is not responding or is responding slowly. The range is from 1 to 1000. If no retransmit value is set with the **radius-server host** command, the value set with the **radius-server retransmit** command is used. • (Optional) For **key** *string*, specify the authentication and encryption key used between the bridge and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key that is used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. To configure the bridge to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The bridge software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove the specified RADIUS server, use the **no radius-server host** *hostname* | *ip-address* command in global configuration mode.

This example shows how to configure one RADIUS server for authentication and another for accounting:

```
bridge(config)# radius-server host 172.29.36.49 auth-port 1612 key rad1
bridge(config)# radius-server host 172.20.36.50 acct-port 1618 key rad2
```

This example shows how to configure *host1* as the RADIUS server using the default ports for both authentication and accounting:

```
bridge(config)# radius-server host host1
```

Configuring RADIUS Login Authentication

To configure AAA authentication, define a named list of authentication methods and apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they are performed; the list must be applied to a specific interface before any of the defined authentication methods are performed. The only exception is the default method list (which is named *default*). The default method list is automatically applied to all interfaces except those for which a named method list is explicitly defined.

A method list describes the sequence and authentication methods to be queried to authenticate a user (in this case, a non-root bridge). Designate one or more security protocols to be used for authentication, to ensure a backup system for authentication if the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle; that is, if the security server or local username database responds by denying the user access—the authentication process stops, and no further authentication methods are attempted.

To configure login authentication, follow these required steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.
Step 3	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]	Creates a login authentication method list. • To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces. For more information on list names, click this link: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fsaaa/scfathen.htm#xtocid2 • For *method1...*, specify the actual method that the authentication algorithm tries. The additional defined methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • **line**—Use the line password for authentication. You must define a line password before you can use this authentication method. Use the **password** *password* line configuration command. • **local**—Use the local username database for authentication. You must enter username information in the database. Use the **username** *password* command in global configuration mode. • **radius**—Use RADIUS authentication. You must configure the RADIUS server before you can use this authentication method. For more information, see the "Identifying the RADIUS Server Host" section.
Step 4	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Enters line configuration mode, and configures the lines to apply the authentication list.

	Command	Purpose
Step 5	**login authentication** {**default** | *list-name*}	Applies the authentication list to a line or set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 6	**radius-server attribute 32 include-in-access-req format %h**	Configures the device to send its system name in the NAS_ID attribute for authentication.
Step 7	**end**	Returns to privileged EXEC mode.
Step 8	**show running-config**	Verifies your entries.
Step 9	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** command in global configuration mode. To disable AAA authentication, use the **no aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] command in global configuration mode. To disable RADIUS authentication for logins or to return to the default value, use the **no login authentication** {**default** | *list-name*} command in line configuration mode.

Defining AAA Server Groups

Configure the bridge to use AAA server groups to group existing server hosts for authentication. Select a subset of the configured server hosts, and use them for a particular service. The server group is used with a global server-host list, which lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server if each entry has a unique identifier (the combination of the IP address and UDP port number), allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. If you configure two different host entries on the same RADIUS server for the same service (such as accounting), the second configured host entry acts as a failover backup to the first one.

Use the **server** group server configuration command to associate a particular server with a defined group server. To identify the server by its IP address or to identify multiple host instances or entries, use the optional **auth-port** and **acct-port** keywords.

To define the AAA server group and associate a particular RADIUS server with it, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.

	Command	Purpose
Step 3	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]	Specifies the IP address or hostname of the remote RADIUS server host. • (Optional) For **auth-port** *port-number*, specify the UDP destination port for authentication requests. • (Optional) For **acct-port** *port-number*, specify the UDP destination port for accounting requests. • (Optional) For **timeout** *seconds*, specify the time interval that the bridge waits for the RADIUS server to reply before retransmitting. The range is 1 to 1000. This setting overrides the interval set using the **radius-server timeout** command. If no timeout is set with the **radius-server host** command, the time interval set with the **radius-server timeout** command is used. • (Optional) For **retransmit** *retries*, specify the number of times a RADIUS request is resent to a server if that server is not responding or is responding slowly. The range is from 1 to 1000. If no retransmit value is set with the **radius-server host** command, the value set with the **radius-server retransmit** command is used. • (Optional) For **key** *string*, specify the authentication and encryption key used between the bridge and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key that is used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. To configure the bridge to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The bridge software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.
Step 4	**aaa group server radius** *group-name*	Defines the AAA server-group with a group name. This command puts the bridge in a server group configuration mode.
Step 5	**server** *ip-address*	Associates a particular RADIUS server with the defined server group. Repeat this step for each RADIUS server in the AAA server group. **Note** Each server in the group must be previously defined in Step 2.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
Step 9		Enables RADIUS login authentication. See the "Configuring RADIUS Login Authentication" section on page 4-24.

To remove the specified RADIUS server, use the **no radius-server host** *hostname* | *ip-address* command in global configuration mode. To remove a server group from the configuration list, use the **no aaa group server radius** *group-name* command in global configuration mode. To remove the IP address of a RADIUS server, use the **no server** *ip-address* command in server group configuration mode.

In this example, the bridge is configured to recognize two different RADIUS group servers (*group1* and *group2*). Group1 has two different host entries on the same RADIUS server configured for the same services. The second host entry acts as a failover backup to the first entry.

```
bridge(config)# aaa new-model
bridge(config)# radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
bridge(config)# radius-server host 172.10.0.1 auth-port 1645 acct-port 1646
bridge(config)# aaa group server radius group1
bridge(config-sg-radius)# server 172.20.0.1 auth-port 1000 acct-port 1001
bridge(config-sg-radius)# exit
bridge(config)# aaa group server radius group2
bridge(config-sg-radius)# server 172.20.0.1 auth-port 2000 acct-port 2001
bridge(config-sg-radius)# exit
```

Configuring RADIUS Authorization for User Privileged Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the bridge uses information retrieved from the user profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** command in global configuration mode with the **radius** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec radius local** command sets these authorization parameters:

- Use RADIUS for privileged EXEC access authorization if authentication was performed by using RADIUS.
- Use the local database if authentication was not performed by using RADIUS.

Note Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

To specify RADIUS authorization for privileged EXEC access and network services, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa authorization network radius**	Configures the bridge for user RADIUS authorization for all network-related service requests.
Step 3	**aaa authorization exec radius**	Configures the bridge for user RADIUS authorization and determines if the user has privileged EXEC access. The **exec** keyword might return user profile information (such as **autocommand** information).
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* command in global configuration mode.

Starting RADIUS Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the bridge reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

To enable RADIUS accounting for each Cisco IOS privilege level and for network services, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting network start-stop radius**	Enables RADIUS accounting for all network-related service requests.
Step 3	**ip radius source-interface bvi1**	Configures the bridge to send its Bridge-Group Virtual Interface (BVI) IP address in the NAS_IP_ADDRESS attribute for accounting records.
Step 4	**aaa accounting update periodic** *minutes*	Enters an accounting update interval in minutes.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable accounting, use the **no aaa accounting** {**network** | **exec**} {**start-stop**} *method1*... command in global configuration mode.

Configuring Settings for All RADIUS Servers

To configure global communication settings between the bridge and all RADIUS servers, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server key** *string*	Specifies the shared secret text string to be used between the bridge and all RADIUS servers. **Note** The key is a text string that must match the encryption key that is used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 3	**radius-server retransmit** *retries*	Specifies the number of times that the bridge sends each RADIUS request to the server before giving up. The default is 3; the range 1 to 1000.
Step 4	**radius-server timeout** *seconds*	Specifies the number of seconds that a bridge waits for a reply to a RADIUS request before resending the request. The default is 5; the range is from 1 to 1000.

	Command	Purpose
Step 5	**radius-server deadtime** *minutes*	Use this command to cause the Cisco IOS software to mark as "dead" any RADIUS servers that fail to respond to authentication requests, thus avoiding the wait for the request to time out before trying the next configured server. A RADIUS server marked as dead is skipped by additional requests for the duration of minutes that you specify. **Note** If you set up more than one RADIUS server, you must configure the RADIUS server deadtime for optimal performance.
Step 6	**radius-server attribute 32 include-in-access-req format %h**	Configures the bridge to send its system name in the NAS_ID attribute for authentication.
Step 7	**end**	Returns to privileged EXEC mode.
Step 8	**show running-config**	Verifies your settings.
Step 9	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default setting for the retransmit, timeout, and deadtime, use the **no** forms of these commands.

Configuring the Bridge to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the bridge and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option by using the format recommended in the specification. Cisco's vendor ID is 9, and the supported option has vendor type 1, which is named *cisco-avpair*. The value is a string with this format:

```
protocol : attribute sep value *
```

Protocol is a value of the Cisco protocol attribute for a particular type of authorization. *Attribute* and *value* are an appropriate AV pair defined in the Cisco TACACS+ specification, and *sep* is = for mandatory attributes and the asterisk (*) for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair activates Cisco's *multiple named ip address pools* feature during IP authorization (during Point-to-Point Protocol IP Control Protocol (PPP IPCP) address assignment):

```
cisco-avpair= ″ip:addr-pool=first“
```

The following example shows how to provide a user logging in from a bridge with immediate access to privileged EXEC commands:

```
cisco-avpair= ″shell:priv-lvl=15“
```

Other vendors have their own unique vendor IDs, options, and associated VSAs. For more information about vendor IDs and VSAs, see RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

To configure the bridge to recognize and use VSAs, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server vsa send** [**accounting** | **authentication**]	Enables the bridge to recognize and use VSAs as defined by RADIUS IETF attribute 26. • (Optional) Use the **accounting** keyword to limit the set of recognized VSAs to only accounting attributes. • (Optional) Use the **authentication** keyword to limit the set of recognized VSAs to only authentication attributes. If you enter this command without keywords, both accounting and authentication VSAs are used.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your settings.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

For a complete list of RADIUS attributes or more information about VSA 26, see the "RADIUS Attributes" appendix in the *Cisco IOS Security Configuration Guide for Release 12.2*.

Configuring the Bridge for Vendor-Proprietary RADIUS Server Communication

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the bridge and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host that is running the RADIUS server daemon and the secret text string that it shares with the bridge. You specify the RADIUS host and secret text string by using the **radius-server** command in global configuration mode.

To specify a vendor-proprietary RADIUS server host and a shared secret text string, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*hostname* | *ip-address*} **non-standard**	Specifies the IP address or hostname of the remote RADIUS server host and identify that it is using a vendor-proprietary implementation of RADIUS.

	Command	Purpose
Step 3	**radius-server key** *string*	Specifies the shared secret text string used between the bridge and the vendor-proprietary RADIUS server. The bridge and the RADIUS server use this text string to encrypt passwords and exchange responses. **Note** The key is a text string that must match the encryption key that is used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your settings.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete the vendor-proprietary RADIUS host, use the **no radius-server host** {*hostname* | *ip-address*} **non-standard** command in global configuration mode. To disable the key, use the **no radius-server key** command in global configuration mode.

This example shows how to specify a vendor-proprietary RADIUS host and a secret key of *rad124* between the bridge and the server:

```
bridge(config)# radius-server host 172.20.30.15 nonstandard
bridge(config)# radius-server key rad124
```

Displaying the RADIUS Configuration

To display the RADIUS configuration, use the **show running-config** command.in privileged EXEC mode:

Controlling WMIC Access with TACACS+

This section describes how to control administrator access to the WMIC using Terminal Access Controller Access Control System Plus (TACACS+).

TACACS+ provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference for Release 12.2*.

Understanding TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your bridge. Unlike RADIUS, TACACS+ does not authenticate non-root bridges that are associated to the root device.

TACACS+ services are maintained in a database on a TACACS+ daemon, which, typically, is running on a UNIX or Windows NT workstation. You should have access to and should configure a TACACS+ server before you configure TACACS+ features on your WMIC.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, or accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

Administered through the AAA security services, TACACS+ can provide these services:

- Authentication—Provides complete control of authentication of administrators through login and password dialog, challenge and response, and messaging support.

  The authentication facility can conduct a dialog with the administrator (for example, after a username and password are provided, to challenge a user with several questions, such as home address, mother's maiden name, service type, and social security number). The TACACS+ authentication service can also send messages to administrator screens. For example, a message could notify administrators that their passwords must be changed because of the company's password aging policy.

- Authorization—Provides "fine-grained" control over administrator capabilities for the duration of the administrator's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on the commands that an administrator can execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track administrator activity for a security audit or to provide information for user billing. Accounting records include administrator identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the WMIC and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between the WMIC and the TACACS+ daemon are encrypted.

Your system must be running the TACACS+ daemon software to use TACACS+ on your WMIC.

TACACS+ Operation

When an administrator attempts a simple ASCII login by authenticating to a WMIC using TACACS+, this process occurs:

1. When the connection is established, the WMIC contacts the TACACS+ daemon to obtain a username prompt, which is then displayed to the administrator. The administrator enters a username; the WMIC then contacts the TACACS+ daemon to obtain a password prompt. The WMIC displays the password prompt to the administrator, the administrator enters a password, and the password is then sent to the TACACS+ daemon.

   TACACS+ allows a conversation between the daemon and the administrator until the daemon receives enough information to authenticate the administrator. The daemon prompts for a username and password combination, but can include other items, such as the user's mother's maiden name.

2. The WMIC eventually receives one of these responses from the TACACS+ daemon:
   - ACCEPT—The administrator is authenticated, and service can begin. If the WMIC is configured to require authorization, authorization begins at this time.
   - REJECT—The administrator is not authenticated. The administrator can be denied access or is prompted to retry the login sequence, depending on the TACACS+ daemon.
   - ERROR—An error occurred at some time during authentication with the daemon or in the network connection between the daemon and the WMIC. If an ERROR response is received, the WMIC typically tries to use an alternative method for authenticating the administrator.
   - CONTINUE—The administrator is prompted for additional authentication information.

   After authentication, the administrator attempts authorization if authorization has been enabled on the WMIC. Administrators must successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the TACACS+ daemon is again contacted, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response contains data in the form of attributes that direct the EXEC or NETWORK session for that administrator, determining the services that the administrator can access:
   - Telnet, rlogin, or privileged EXEC services
   - Connection parameters, including the host or client IP address, access list, and administrator timeouts

Default TACACS+ Configuration

TACACS+ and AAA are disabled by default.

To prevent a lapse in security, you cannot configure TACACS+ through a network management application.When enabled, TACACS+ can authenticate administrators accessing the WMIC through the CLI.

Configuring TACACS+ Login Authentication

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they are performed; the list must be applied to a specific interface before any of the defined authentication methods are performed. The only exception is the default method list (which is named *default*).

The default method list is automatically applied to all interfaces except those for which a named method list is explicitly defined. A defined method list overrides the default method list.

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, to ensure a backup system for authentication if the initial method fails. The software uses the first method listed to authenticate users; if that method fails, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails—that is, the security server or local username database responds by denying the user access— the authentication process stops, and no further authentication methods are attempted.

Identifying the TACACS+ Server Host and Setting the Authentication Key

You can configure the WMIC to use a single server or to use AAA server groups to group existing server hosts for authentication. You can group servers to select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list and contains the list of IP addresses of the selected server hosts.

To identify the IP host or host maintaining TACACS+ server and optionally set the encryption key, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** *hostname* [**port** *integer*] [**timeout** *integer*] [**key** *string*]	Identifies the IP host or hosts maintaining a TACACS+ server. Enter this command multiple times to create a list of preferred hosts. The software searches for hosts in the order in which you specify them. • For *hostname*, specify the name or IP address of the host. • (Optional) For **port** *integer*, specify a server port number. The default is port 49. The range is from 1 to 65535. • (Optional) For **timeout** *integer*, specify a time, in seconds, that the WMIC waits for a response from the daemon before it times out and declares an error. The default is 5. The range is from 1 to 1000. • (Optional) For **key** *string*, specify the encryption key for encrypting and decrypting all traffic between the WMIC and the TACACS+ daemon. For encryption to be successful, you must configure the same key on the TACACS+ daemon.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**aaa group server tacacs+** *group-name*	(Optional) Defines the AAA server-group with a group name. This command puts the WMIC in a server group subconfiguration mode.

	Command	Purpose
Step 5	**server** *ip-address*	(Optional) Associate a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group. Each server in the group must be previously defined in Step 2.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show tacacs**	Verifies your entries.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove the specified TACACS+ server name or address, use the **no tacacs-server host** *hostname* command in global configuration mode. To remove a server group from the configuration list, use the **no aaa group server tacacs+** *group-name* command in global configuration mode. To remove the IP address of a TACACS+ server, use the **no server ip-address** server group subconfiguration command.

Configuring TACACS+ Login Authentication

To configure AAA authentication, define a named list of authentication methods, and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they are performed; the method must be applied to a specific interface before any of the defined authentication methods can be performed. The only exception is the default method list (which is named *default*). The default method list is automatically applied to all interfaces except those for which a named method list is explicitly defined. A defined method list overrides the default method list.

A method list describes the sequence and authentication methods to be queried to authenticate an administrator. You can designate one or more security protocols to be used for authentication, to ensure a backup system for authentication if the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—that is, if the security server or local username database responds by denying the administrator access—the authentication process stops, and no further authentication methods are attempted.

To configure login authentication, follow these required steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.

	Command	Purpose
Step 3	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]	Create a login authentication method list. • To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces. • For *list-name*, specify a character string as the name the list you are creating. • For *method1...*, specify the actual method that the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • **local**—Use the local username database for authentication. You must enter username information into the database. Use the **username** *password* global configuration command. • **tacacs+**—Use TACACS+ authentication. You must configure the TACACS+ server before you can use this authentication method.
Step 4	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Enters line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 5	**login authentication** {**default** | *list-name*}	Apply the authentication list to a line or a set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** command in global configuration mode. To disable AAA authentication, use the **no aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] command in global configuration mode. To either disable TACACS+ authentication for logins or to return to the default value, use the **no login authentication** {**default** | *list-name*} command in line configuration mode.

Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services

AAA authorization limits the services that are available to a user. When AAA authorization is enabled, the WMIC uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** command in global configuration mode with the **tacacs+** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec tacacs+ local** command sets these authorization parameters:

- Use TACACS+ for privileged EXEC access authorization if authentication was performed by using TACACS+.
- Use the local database if authentication was not performed by using TACACS+.

Note Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

To specify TACACS+ authorization for privileged EXEC access and network services, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa authorization network tacacs+**	Configures the WMIC for user TACACS+ authorization for all network-related service requests.
Step 3	**aaa authorization exec tacacs+**	Configures the WMIC for user TACACS+ authorization to determine whether the user has privileged EXEC access. The **exec** keyword might return user profile information (such as **autocommand** information).
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* command in global configuration mode.

Starting TACACS+ Accounting

The AAA accounting feature tracks the services that administrators are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the WMIC reports administrator activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs, and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

To enable TACACS+ accounting for each Cisco IOS privilege level and for network services, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting network start-stop tacacs+**	Enables TACACS+ accounting for all network-related service requests.
Step 3	**aaa accounting exec start-stop tacacs+**	Enables TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 4	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable accounting, use the **no aaa accounting** {**network** | **exec**} {**start-stop**} *method1*... command in global configuration mode.

Displaying the TACACS+ Configuration

To display TACACS+ server statistics, use the **show tacacs** command in privileged EXEC mode.

Configuring the WMIC for Local Authentication and Authorization

You can configure AAA to operate without a server by setting the WMIC to implement AAA in local mode. The WMIC then handles authentication and authorization. No accounting is available in this configuration.

To configure the WMIC for local AAA, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.
Step 3	**aaa authentication login default local**	Sets the login authentication to use the local username database. The **default** keyword applies the local user database authentication to all interfaces.
Step 4	**aaa authorization exec local**	Configures user AAA authorization to check the local database to determine whether the user is allowed to run an EXEC shell.
Step 5	**aaa authorization network local**	Configures user AAA authorization for all network-related service requests.

	Command	Purpose
Step 6	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}	Enters the local database, and establish a username-based authentication system. Repeat this command for each user. • For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed. • (Optional) For *level*, specify the privilege level the user has after gaining access. The range is from 0 to 15. Level 15 gives privileged EXEC mode access. Level 0 gives user EXEC mode access. • For *encryption-type*, enter **0** to specify that an unencrypted password follows. Enter **7** to specify that a hidden password follows. • For *password*, specify the password the user must enter to gain access to the WMIC. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
Step 7	**end**	Returns to privileged EXEC mode.
Step 8	**show running-config**	Verifies your entries.
Step 9	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** command in global configuration mode. To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* command in global configuration mode.

Configuring the WMIC for Secure Shell

This section describes how to configure the Secure Shell (SSH) feature.

Note For complete syntax and usage information for the commands used in this section, see the "Secure Shell Commands" section in the *Cisco IOS Security Command Reference for Release 12.2*.

Understanding SSH

SSH is a protocol that provides a secure, remote connection to a Layer 2 or a Layer 3 device. There are two versions of SSH: SSH version 1 and SSH version 2. Cisco IOS release 12.3(8)JK supports only SSH version 1.

SSH provides greater security for remote connections than Telnet provides. When a device is authenticated, SSH provides strong encryption. The SSH feature has an SSH server and an SSH integrated client. The client supports these user authentication methods:

- RADIUS (for more information, see the "Controlling WMIC Access with RADIUS" section on page 4-21)
- Local authentication and authorization (for more information, see the "Configuring the WMIC for Local Authentication and Authorization" section on page 4-38)

For more information about SSH, see the "Configuring Secure Shell" section in the *Cisco IOS Security Configuration Guide for Release 12.2*.

Note The SSH feature in Cisco IOS release 12.3(8)JK does not support IP Security (IPSec).

Configuring SSH

Before you configure SSH, download the crypto software image from Cisco.com. For information about configuring SSH and displaying SSH settings, see the "Configuring Secure Shell" section in the *Cisco IOS Security Configuration Guide for Release 12.2*.

Managing Aironet Extensions

The WMIC uses Cisco Aironet 802.11 extensions to detect the capabilities of Cisco client devices and to support features that require specific interaction between the WMIC and associated client devices. The Aironet extensions can be deactivated only in the root access point mode. Since workgroup bridge, root device, and non-root bridge are Cisco-specific modes, they always use the Aironet extensions.

Aironet extensions must be enabled to support the following features:

- Load balancing—The WMIC uses Aironet extensions to direct client devices to an access point that provides the best connection to the network, based on such factors as number of users, bit error rates, and signal strength.
- Message Integrity Check (MIC)—MIC is an additional WEP security feature that prevents attacks on encrypted packets called *bit-flip attacks*. The MIC, implemented on both the WMIC and all associated client devices, adds a few bytes to each packet to make the packets tamper-proof.
- Temporal Key Integrity Protocol (TKIP)—TKIP, also known as *WEP key hashing*, is an additional WEP security feature that defends against an attack on WEP in which the intruder uses an unencrypted segment called the *initialization vector* (IV) in encrypted packets to calculate the WEP key.
- Limiting the power level on associated client devices—When a client device associates to the WMIC, the WMIC sends the maximum allowed power level setting to the client.

To disable the Aironet extensions, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface dot11radio 0**	Enters interface configuration mode for the radio interface.
Step 3	**station-role root ap-only**	Enters the station role. Root enables the access point mode.
Step 4	**no dot11 extension aironet**	Enters the **extension aironet** command to disable extensions.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

If you change the radio to a role that requires Aironet extensions, the Aironet extensions are enabled automatically:

```
wmic1(config)#int dot 0
```

```
wmic1(config-if)#station-role root
Selected role requires Cisco Aironet Extension enabled.
Enabled Cisco Aironet Extension.
```

If you try to change the Aironet extensions without setting the radio to the proper role, an error message displays:

```
wmic1(config-if)#
wmic1(config-if)#no dot11 extension aironet
Aironet Extension is always enabled in Bridge or WGB mode.
```

Managing the System Time and Date

You can manage the system time and date on your WMIC automatically, by using the Network Time Protocol (NTP), or manually, by setting the time and date on the WMIC.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.2*.

Understanding the System Clock

The heart of the time service is the system clock. This clock runs from the moment that the system starts up. The system clock keeps track of the date and time.

The system clock can then be set from these sources:

- Network Time Protocol
- Manual configuration

The system clock determines time internally, based on Universal Time Coordinated (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight saving time) so that the time is correctly displayed for the local time zone.

The system clock keeps track of whether the time is *authoritative* or not (that is, whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time is available only for display purposes and is not redistributed. For configuration information, see the "Configuring Time and Date Manually" section on page 4-43.

Understanding Network Time Protocol

The NTP is designed to time-synchronize a network of devices. NTP is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two devices to within a millisecond of each other.

NTP uses the concept of a *stratum* to describe how many NTP hops away a device is from an authoritative time source. A stratum 1 time server has a radio or atomic clock directly attached, a stratum 2 time server receives its time through NTP from a stratum 1 time server, and so on.

A device running NTP automatically chooses as its time source the device with the lowest stratum number with which it communicates through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP compares the time reported by several devices and does not synchronize to a device whose time is significantly different than the others, even if its stratum is lower.

The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP address of all the devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between associated devices. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can simply be configured to send or receive broadcast messages. However, in that case, information flow is one-way only.

The time kept on a device is a critical resource; you should use the security features of NTP to avoid the accidental or malicious setting of an incorrect time. Two mechanisms are available: an access-list-based restriction scheme and an encrypted authentication mechanism.

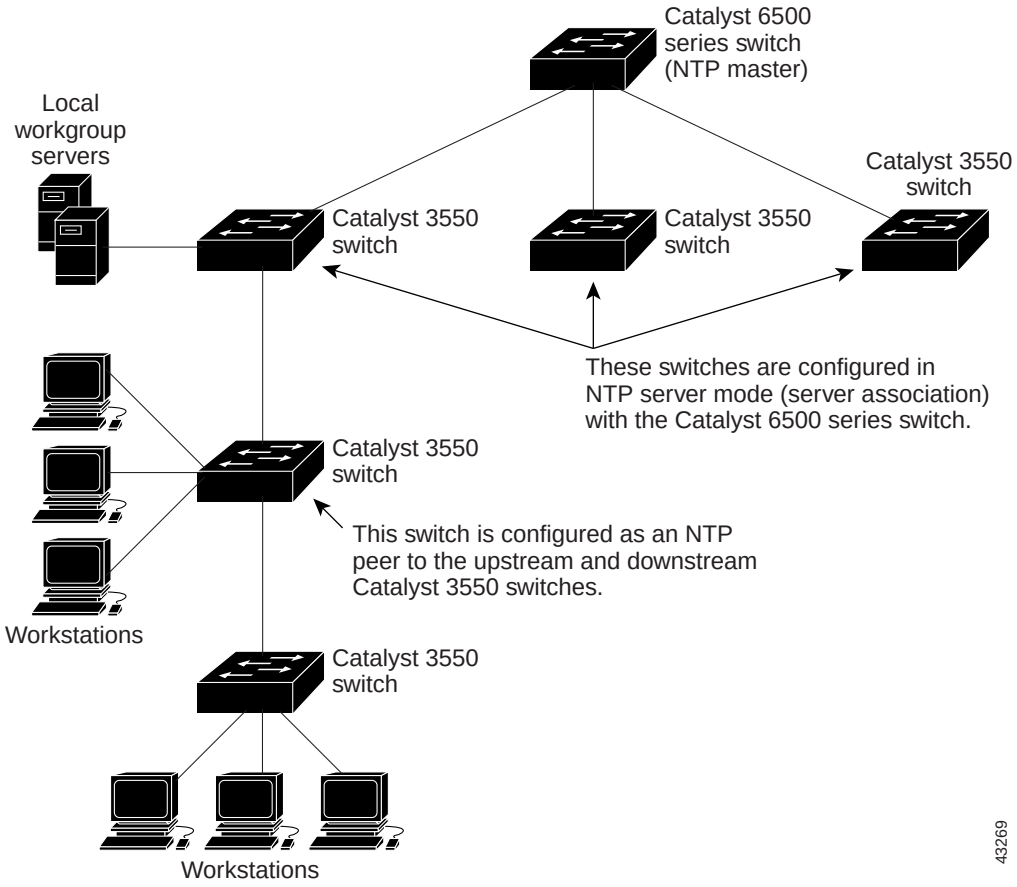
Cisco's implementation of NTP does not support stratum 1 service; it is not possible to connect to a radio or atomic clock. We recommend that the time service for your network be derived from the public NTP servers available on the IP Internet. Figure 4-2 shows a typical network example using NTP.

If the network is isolated from the Internet, Cisco's implementation of NTP allows a device to act as though it is synchronized through NTP, when in fact it has determined the time by using other means. Other devices then synchronize to that device through NTP.

When multiple sources of time are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Several manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. The NTP software allows host systems to be time-synchronized as well.

Figure 4-2 Typical NTP Network Configuration



Configuring Time and Date Manually

If no other source of time is available, you can manually configure the time and date after the system is restarted. The time remains accurate until the next system restart. We recommend that you use manual configuration only as a last resort. If you have an outside source to which the WMIC can synchronize, you do not need to manually set the system clock.

Setting the System Clock

If you have an outside source on the network that provides time services, such as an NTP server, you do not need to manually set the system clock.

To set the system clock, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**clock set** *hh*:*mm*:*ss day month year* or **clock set** *hh*:*mm*:*ss month day year*	Manually set the system clock using one of these formats. • For *hh*:*mm*:*ss*, specify the time in hours (24-hour format), minutes, and seconds. The time specified is relative to the configured time zone. • For *day*, specify the day by date in the month. • For *month*, specify the month by name. • For *year*, specify the year (no abbreviation).
Step 2	**show running-config**	Verifies your entries.
Step 3	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to manually set the system clock to 1:32 p.m. on July 23, 2001:

```
bridge# clock set 13:32:00 23 July 2001
```

Displaying the Time and Date Configuration

To display the time and date configuration, use the **show clock** [**detail**] command in privileged EXEC mode.

The system clock keeps an *authoritative* flag that shows whether the time is authoritative (believed to be accurate). If the system clock has been set by a timing source such as NTP, the flag is set. If the time is not authoritative, the flag is used only for display purposes. Until the clock is authoritative and the *authoritative* flag is set, the flag prevents peers from synchronizing to the clock when the peers' time is invalid.

The symbol that precedes the **show clock** display has this meaning:

- *—Time is not authoritative.
- (blank)—Time is authoritative.
- .—Time is authoritative, but NTP is not synchronized.

Configuring the Time Zone

To manually configure the time zone, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock timezone** *zone hours-offset* [*minutes-offset*]	Sets the time zone. The device keeps internal time in universal time coordinated (UTC), so this command is used only for display purposes and when the time is manually set. • For *zone*, enter the name of the time zone to be displayed when standard time is in effect. The default is UTC. • For *hours-offset*, enter the hours offset from UTC. • (Optional) For *minutes-offset*, enter the minutes offset from UTC.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The *minutes-offset* variable in the **clock timezone** command is available for those areas where a local time zone is only a percentage of an hour different from UTC. For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5, where the 3 means 3 hours and .5 means 50 percent. In this case, the necessary command is **clock timezone AST -3 30**.

To set the time to UTC, use the **no clock timezone** command in global configuration mode.

Configuring Summer Time (Daylight Saving Time)

To configure daylight saving time in areas where it starts and ends on a particular day of the week each year, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock summer-time** *zone* **recurring** [*week day month hh*:*mm week day month hh*:*mm* [*offset*]]	Configures summer time to start and end on the specified days every year. Summer time is disabled by default. If you specify **clock summer-time** *zone* **recurring** without parameters, the summer time rules default to the United States rules. • For *zone*, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. • (Optional) For *week*, specify the week of the month (1 to 5 or **last**). • (Optional) For *day*, specify the day of the week (Sunday, Monday...). • (Optional) For *month*, specify the month (January, February...). • (Optional) For *hh*:*mm*, specify the time (24-hour format) in hours and minutes. • (Optional) For *offset*, specify the number of minutes to add during summer time. The default is 60.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** command specifies when daylight savings time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to daylight savings time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

This example shows how to specify that summer time starts on the first Sunday in April at 02:00 and ends on the last Sunday in October at 02:00:

```
bridge(config)# clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October
2:00
```

Configuring NTP

WMICs do not have a hardware-supported clock, and they cannot function as an NTP master clock to which peers synchronize when an external NTP source is not available. These devices also have no hardware support for a calendar. As a result, the **ntp update-calendar** and the **ntp master** commands are not available.

Default NTP Configuration

Table 4-4 shows the default NTP configuration.

Table 4-4 Default NTP Configuration

Feature	Default Setting
NTP authentication	Disabled. No authentication key is specified.
NTP peer or server associations	None configured.
NTP broadcast service	Disabled; no interface sends or receives NTP broadcast packets.
NTP access restrictions	No access control is specified.
NTP packet source IP address	The source address is determined by the outgoing interface.

NTP is disabled by default.

Configuring NTP Authentication

This procedure must be coordinated with the administrator of the NTP server; the information you configure in this procedure must be matched by the servers that the WMIC uses to synchronize its time to the NTP server.

To authenticate the associations (communications between devices running NTP that provide for accurate timekeeping) with other devices, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp authenticate**	Enables the NTP authentication feature, which is disabled by default.
Step 3	**ntp authentication-key** *number* **md5** *value*	Defines the authentication keys. By default, none are defined. • For *number*, specify a key number. The range is from 1 to 4294967295. • **md5** specifies that message authentication support is provided by using the message digest algorithm 5 (MD5). • For *value*, enter an arbitrary string of up to eight characters for the key. The WMIC does not synchronize to a device unless both it and the device have an authentication key, and the key number is specified by the **ntp trusted-key** *key-number* command.

	Command	Purpose
Step 4	**ntp trusted-key** *key-number*	Specifies one or more key numbers (defined in Step 3) that a peer NTP device must provide in its NTP packets for this WMIC to synchronize to it. By default, no trusted keys are defined. For *key-number*, specify the key defined in Step 3. This command provides protection against accidentally synchronizing the WMIC to any untrusted device.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable NTP authentication, use the **no ntp authenticate** command in global configuration mode. To remove an authentication key, use the **no ntp authentication-key** *number* command in global configuration mode. To disable authentication of the identity of a device, use the **no ntp trusted-key** *key-number* command in global configuration mode.

This example shows how to configure the WMIC to synchronize only to devices providing authentication key 42 in the device's NTP packets:

```
bridge(config)# ntp authenticate
bridge(config)# ntp authentication-key 42 md5 aNiceKey
bridge(config)# ntp trusted-key 42
```

Configuring NTP Associations

An NTP association can be a peer association (the WMIC can either synchronize to the other device or allow the other device to synchronize to it), or it can be a server association (the WMIC synchronizes to the other device; the device does not synchronize to the WMIC).

To form an NTP association with another device, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp peer** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]	Configures the WMIC system clock to synchronize a peer or to be synchronized by a peer (peer association).
	or	or
	ntp server *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]	Configures the WMIC system clock to be synchronized by a time server (server association). No peer or server associations are defined by default. • For *ip-address* in a peer association, specify the IP address of the peer that is providing, or that is being provided, the clock synchronization. For a server association, specify the IP address of the time server that is providing the clock synchronization. • (Optional) For *number*, specify the NTP version number. The range is from 1 to 3. By default, version 3 is selected. • (Optional) For *keyid*, enter the authentication key defined with the **ntp authentication-key** command. • (Optional) For *interface*, specify the interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • (Optional) Enter the **prefer** keyword to make this peer or server the preferred one that provides synchronization. This keyword reduces switching between peers and servers.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

You need to configure only one end of an association; the other device can automatically establish the association. If you are using the default NTP version (version 3), but NTP synchronization does not occur, try using NTP version 2. Many NTP servers on the Internet run version 2.

To remove a peer or server association, use the **no ntp peer** *ip-address* or the **no ntp server** *ip-address* command in global configuration mode.

This example shows how to configure the WMIC to synchronize its system clock with the clock of the peer at IP address 172.16.22.44, using NTP version 2:

```
bridge(config)# ntp server 172.16.22.44 version 2
```

Configuring NTP Broadcast Service

The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP addresses of all the devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between associated devices. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can be configured only to send broadcast messages or to receive broadcast messages. However, the information flow is one-way only.

The WMIC can send or receive NTP broadcast packets on an interface-by-interface basis if there is an NTP broadcast server, such as a router, broadcasting time information on the network. The WMIC can send NTP broadcast packets to a peer so that the peer can synchronize to it. The WMIC can also receive NTP broadcast packets to synchronize its own clock. This section provides procedures for sending and receiving NTP broadcast packets.

To configure the WMIC to send NTP broadcast packets to peers so that they can synchronize their clock to the WMIC, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to send NTP broadcast packets.
Step 3	**ntp broadcast** [**version** *number*] [**key** *keyid*] [*destination-address*]	Enables the interface to send NTP broadcast packets to a peer. By default, this feature is disabled on all interfaces. • (Optional) For *number*, specify the NTP version number. The range is 1 to 3. If you do not specify a version, version 3 is used. • (Optional) For *keyid*, specify the authentication key to use when sending packets to the peer. • (Optional) For *destination-address*, specify the IP address of the peer that is synchronizing its clock to this WMIC.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable the interface from sending NTP broadcast packets, use the **no ntp broadcast** interface configuration command.

This example shows how to configure an interface to send NTP version 2 packets:

```
bridge(config)# interface gigabitethernet0/1
bridge(config-if)# ntp broadcast version 2
```

Now you are ready to configure the connected peers to receive NTP broadcast packets, as described in the next procedure. To configure the WMIC to receive NTP broadcast packets from connected peers, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specifies the interface to receive NTP broadcast packets.
Step 3	**ntp broadcast client**	Enables the interface to receive NTP broadcast packets. By default, no interfaces receive NTP broadcast packets.
Step 4	**exit**	Returns to global configuration mode.
Step 5	**ntp broadcastdelay** *microseconds*	(Optional) Changes the estimated round-trip delay between the WMIC and the NTP broadcast server. The default is 3000 microseconds; the range is from 1 to 999999.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable an interface from receiving NTP broadcast packets, use the **no ntp broadcast client** command in configuration mode. To change the estimated round-trip delay to the default, use the **no ntp broadcastdelay** command in global configuration mode.

This example shows how to configure an interface to receive NTP broadcast packets:

```
bridge(config)# interface gigabitethernet0/1
bridge(config-if)# ntp broadcast client
```

Configuring NTP Access Restrictions

You can control NTP access by using access lists.

Creating an Access Group and Assigning a Basic IP Access List

To control access to NTP services by using access lists, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} *access-list-number*	Creates an access group, and applies a basic IP access list. The keywords have these meanings: • **query-only**—Allows only NTP control queries. • **serve-only**—Allows only time requests. • **serve**—Allows time requests and NTP control queries, but does not allow the WMIC to synchronize to the remote device. • **peer**—Allows time requests and NTP control queries and allows the WMIC to synchronize to the remote device. For *access-list-number*, enter a standard IP access list number from 1 to 99.
Step 3	**access-list** *access-list-number* **permit** *source* [*source-wildcard*]	Creates the access list. • For *access-list-number*, enter the number specified in Step 2. • Enter the **permit** keyword to permit access if the conditions are matched. • For *source*, enter the IP address of the device that is permitted access to the WMIC. • (Optional) For *source-wildcard*, enter the wildcard bits to be applied to the source. **Note** When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The access group keywords are scanned in this order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the WMIC to synchronize itself to a device whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the WMIC to synchronize itself to a device whose address passes the access list criteria.

3. **serve-only**—Allows only time requests from a device whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a device whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first access type is granted. If no access groups are specified, all access types are granted to all devices. If any access groups are specified, only the specified access types are granted.

To remove access control to the WMIC NTP services, use the **no ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} command in global configuration mode.

This example shows how to configure the WMIC to allow itself to synchronize to a peer from access list 99. However, the WMIC restricts access to allow only time requests from access list 42:

```
bridge# configure terminal
bridge(config)# ntp access-group peer 99
bridge(config)# ntp access-group serve-only 42
bridge(config)# access-list 99 permit 172.20.130.5
bridge(config)# access list 42 permit 172.20.130.6
```

Disabling NTP Services on a Specific Interface

NTP services are enabled on all interfaces by default.

To disable NTP packets from being received on an interface, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to disable.
Step 3	**ntp disable**	Disables NTP packets from being received on the interface. By default, all interfaces receive NTP packets.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To reenable receipt of NTP packets on an interface, use the **no ntp disable** command in interface configuration mode.

Configuring the Source IP Address for NTP Packets

When the WMIC sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the **ntp source** command in global configuration mode when you want to use a particular source IP address for all NTP packets. The address is taken from the specified interface. This command is useful if the address on an interface cannot be used as the destination for reply packets.

To configure a specific interface from which the IP source address is to be taken, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp source** *type number*	Specifies the interface type and number from which the IP source address is taken. By default, the source address is determined by the outgoing interface.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The specified interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **ntp peer** or **ntp server** command in global configuration mode as described in the "Configuring NTP Associations" section on page 4-48.

Displaying the NTP Configuration

To display NTP information, use the following commands in privileged EXEC mode:

- **show ntp associations** [**detail**]
- **show ntp status**

For detailed information about the fields in these displays, see the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.1*.

If summer time in your area does not follow a recurring pattern (configure the exact date and time of the next summer time events), follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock summer-time** *zone* **date** [*month date year hh*:*mm month date year hh*:*mm* [*offset*]] or **clock summer-time** *zone* **date** [*date month year hh*:*mm date month year hh*:*mm* [*offset*]]	Configures summer time to start on the first date and end on the second date. Summer time is disabled by default. • For *zone*, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. • (Optional) For *week*, specify the week of the month (1 to 5 or **last**). • (Optional) For *day*, specify the day of the week (Sunday, Monday...). • (Optional) For *month*, specify the month (January, February...). • (Optional) For *hh*:*mm*, specify the time (24-hour format) in hours and minutes. • (Optional) For *offset*, specify the number of minutes to add during summer time. The default is 60.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

To disable summer time, use the **no clock summer-time** command in global configuration mode.

This example shows how to set summer time to start on October 12, 2005, at 02:00, and to end on April 26, 2006, at 02:00:

```
bridge(config)# clock summer-time pdt date 12 October 2000 2:00 26 April 2001 2:00
```