



## USB Storage

---

The USB Storage feature enables certain models of Cisco routers to support USB flash modules and with SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) to provide secure access to a router.

USB eTokens provides secure configuration distribution and allows users to store Virtual Private Network (VPN) credentials for deployment. USB flash drives allow users to store images and configurations external to the router.

### Feature History for USB Storage

Release	Modification
12.3(14)T	This feature was introduced.

### Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

## Contents

- [Prerequisites for USB Storage, page 2](#)
- [Restrictions for USB Storage, page 2](#)
- [Information About USB Storage, page 2](#)
- [How to Set Up and Use USB Modules on Cisco Routers, page 4](#)
- [Configuration Examples for Secure Token Support, page 14](#)
- [Additional References, page 16](#)
- [Command Reference, page 17](#)



---

Corporate Headquarters:  
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Copyright © 2005 Cisco Systems, Inc. All rights reserved.

## Prerequisites for USB Storage

Before you can use a USB Flash module or an eToken, you should have the following system requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, or a Cisco 3800 series router
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms
- A Cisco supported USB flash or USB eToken
- A k9 image is required for USB eToken support. (However, USB flash support is available in all images.)

## Restrictions for USB Storage

- USB eToken support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports on the router chassis.
- You cannot boot an image from an eToken or a USB flash. (However, you can boot a configuration from both an eToken and flash.)

## Information About USB Storage

To use a USB flash module and a secure eToken on your router, you should understand the following concepts:

- [Roles of the USB eToken and the USB Flash, page 2](#)
- [Benefits of USB Storage, page 4](#)

## Roles of the USB eToken and the USB Flash

Both USB eTokens and USB flash modules can be used to store files (such as router configurations). The following sections discuss how each device functions and describe the differences between each device:

- [How a USB eToken Works, page 2](#)
- [How a USB Flash Works, page 3](#)
- [Functionality Differences Between an eToken and a USB Flash, page 3](#)

## How a USB eToken Works

A SmartCard is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A SmartCard eToken is a SmartCard with a USB interface. The eToken can securely store any type of file within its available storage space (32KB). Configuration files that are stored on the eToken can be encrypted and accessed only via a user PIN. The router will not load the configuration file unless the proper PIN has been configured for secure deployment of router configuration files.

After you plug the eToken into the router, you must log into the eToken; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts before future logins are refused (default: 15 attempts). For more information on accessing and configuring the eToken, see the section “[Accessing and Setting Up the eToken.](#)”

After you have successfully logged into the eToken, you can copy files from the router on to the eToken via the **copy** command. By default, after the eToken is removed from the router, all associated RSA keys are removed; IPSec tunnels are not torn down until the next Internet Key Exchange (IKE) negotiation period. (To change the default behavior and configure a specified length of time before the IPSec tunnels are torn down, issue the **crypto pki token removal timeout** command.)

For more information about the eToken by Aladdin Knowledge Systems, see the Aladdin website at <http://www.aladdin.com/etoken/cisco/>.

## How a USB Flash Works

A Cisco USB flash module allows you to store and deploy router configurations and Cisco IOS software images. Cisco USB flash modules are available in 64MB, 128 MB, and 256MB versions.



Note

The USB flash is not a replacement for the router compact flash, which must be present for the router to boot.

After you plug the USB flash module into the router, the router will automatically begin to boot the configuration file if the start-up configuration contains the **boot config** command to specify the new configuration located on the USB flash device; for example **boot config usbflash0: new-config**.

## Functionality Differences Between an eToken and a USB Flash

Both eTokens and USB flash provide users with secondary storage; however, each device has its own benefits and limitations. To help determine which device better suits your needs, [Table 1](#) highlights the functionality differences between the eToken and the USB flash.

**Table 1**      *Functionality Differences Between an eToken and a USB Flash*

Function	USB eToken	USB Flash
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and router configurations from the eToken to the router.	Used to store and deploy router configurations and images from the USB Flash to the router.
Storage Size	32KB	<ul style="list-style-type: none"> <li>• 64MB</li> <li>• 128MB</li> <li>• 256MB</li> </ul>
File Types	<ul style="list-style-type: none"> <li>• Typically used to store digital certificates, preshared keys, and router configurations for IPSec VPNs.</li> <li>• eTokens cannot store Cisco IOS images.</li> </ul>	Stores a file type that might be stored on a compact flash.

**Table 1**      *Functionality Differences Between an eToken and a USB Flash (Continued)*

Function	USB eToken	USB Flash
Security	<ul style="list-style-type: none"> <li>Files can be encrypted and accessed only with a user PIN.</li> <li>Files can also be stored in a nonsecure format.</li> </ul>	Files can be stored only in a nonsecure format.
Boot Configurations	<ul style="list-style-type: none"> <li>The router can use the configuration stored in the eToken during boot time</li> <li>The router can use the secondary configuration stored in the eToken during boot time. (A secondary configuration allows users to load their IPSec configuration.)</li> </ul>	<ul style="list-style-type: none"> <li>Configuration file can be automatically transferred from the USB Flash to the router if the <b>boot config</b> command is issued (for example, <b>boot config usbflash0: new-config</b>).</li> </ul>

## Benefits of USB Storage

USB flash drive and USB eToken support on a Cisco router provides the following application benefits:

### Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

An Aladdin eToken can use SmartCard technology to store a digital certificate and configuration for IPSec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPSec tunnel. (Because a router can initiate multiple IPSec tunnels, the eToken can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

### PIN Configuration for Secure File Deployment

An Aladdin eToken can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

### Touchless or Low Touch Configuration

Both the eToken and USB Flash can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, both devices can store a bootstrap configuration that the router can use to boot from after the eToken or USB Flash has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

## How to Set Up and Use USB Modules on Cisco Routers

This section contains the following procedures that allow you to configure a router to support USB modules:

- [Storing the Configuration on an External USB Flash Drive or eToken, page 5](#)
- [Accessing and Setting Up the eToken, page 5](#)
- [Troubleshooting USB Flash Drives and eTokens, page 9](#)

## Storing the Configuration on an External USB Flash Drive or eToken

Use the following task to store the configuration file in the USB flash drive module or in an eToken.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot config {usbflash[0-9]:filename | usbtoken[0-9]:filename}**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal</p>	Enters global configuration mode.
Step 3	<pre>boot config {usbflash[0-9]:filename   usbtoken[0-9]:filename}</pre> <p><b>Example:</b> Router(config)# boot config usbflash0: </p>	Specifies that the startup configuration file is stored in a USB Flash drive or secure eToken. <p><b>Note</b> If a USB flash drive is used, the router will boot a boot helper from <b>flash:</b>. The boot helper is a Cisco IOS image that resides in <b>flash:</b>. The Cisco IOS image that is used must be USB-aware.</p>

## Accessing and Setting Up the eToken

After you have inserted the eToken into the Cisco router, you must log into the eToken as shown in the following task:

- [Logging Into the eToken, page 6](#) (required)

After you have logged into the eToken, you can perform administrative tasks, such as changing the user PIN and copying files from the router to the eToken, as shown in the following task:

- [Setting Administrative Functions on the eToken, page 7](#) (optional)

## Use of RSA Keys with an eToken

- RSA keys are loaded after the eToken is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted eToken. Regenerated keys should be stored in the same location that the original RSA key was generated.

## Logging Into the eToken

Use this task to log into an eToken manually or automatically.

### Automatic Login

Automatic login allows the router to completely come back up without any user or operator intervention. The PIN is stored in the private configuration, so it is not visible in the startup or running configuration.



#### Note

A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the **copy system:running-config nvram: startup-config** command must be issued to put the hand-generated configuration in the private configuration.

### Manual Login

Manual login can be used when storing a PIN on the router is not desirable. Manual login can be executed with or without privileges, and it will make files and RSA keys on the eToken available to the Cisco IOS software. If a secondary configuration file is configured, it will only be executed with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the eToken to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the router configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the eToken can provide. The eToken can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the router is obtained from the local supplier or drop-shipped to the remote site.

Unlike automatic login, manual login requires that the user know the actual token PIN. However, if the user also has physical access to the eToken, he or she can use Aladdin's Windows-based utilities to copy the RSA keys and secondary config files from the eToken.

## SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]  
or  
**configure terminal**
3. **crypto pki token** *token-name* **user-pin** [*pin*]
4. **exit**
5. **show usbtoken**[0-9];*filename*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> <code>enable</code>	Enables privileged EXEC mode.  • Enter your password if prompted.
Step 2	<code>crypto pki token token-name [admin] login [pin]</code>  <b>Example:</b> Router# <code>crypto pki token usbtokens0 admin login 5678</code>  or  <code>configure terminal</code>  <b>Example:</b> Router# <code>configure terminal</code>	Manually logs into the eToken.  You must specify the <b>admin</b> keyword if later you want to change the user PIN.  or  Puts the router in global configuration mode, which allows you to configure automatic eToken login.
Step 3	<code>crypto pki token token-name user-pin [pin]</code>  <b>Example:</b> Router(config)# <code>crypto pki token usbtokens0 user-pin 1234</code>	(Optional) Creates a PIN that automatically allows the router to log into the USB eToken at router startup.  <b>Note</b> Do not issue this command if you have already set up manual login.
Step 4	<code>exit</code>  <b>Example:</b> Router(config)# <code>exit</code>	Exits global configuration mode.
Step 5	<code>show usbtokens [0-9] : filename</code>  <b>Example:</b> Router#	(Optional) Verifies whether the USB eToken has been logged onto the router.

## Setting Administrative Functions on the eToken

Use this task to change default settings, such as the user PIN and the maximum number of failed on the eToken.

## SUMMARY STEPS

1. `enable`
2. `crypto pki token token-name [admin] change-pin [pin]`
3. `configure terminal`
4. `crypto pki token {token-name | default} removal timeout [minutes]`
5. `crypto pki token {token-name | default} max-retries [number]`
6. `exit`

7. **copy usbflash[0-9]:filename destination-url**
8. **show usbtoken[0-9]:filename**
9. **crypto pki token token-name logout**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<p><b>crypto pki token token-name [admin] change-pin [pin]</b></p> <p><b>Example:</b> Router# crypto pki token usbtoken0 admin change-pin</p>	<p>(Optional) Changes the user PIN number on the USB eToken.</p> <ul style="list-style-type: none"> <li>• If the PIN is not changed, the default PIN—1234567890—will be used.</li> </ul> <p><b>Note</b> After the PIN has been changed, you must reset the login failure count to zero (via the <b>crypto pki token max-retries</b> command). The maximum number of allowable login failures is set (by default) to 15.</p>
Step 3	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 4	<p><b>crypto pki token {token-name   default} removal timeout [seconds]</b></p> <p><b>Example:</b> Router(config)# crypto pki token usbtoken0 removal timeout 60</p>	<p>(Optional) Sets the time interval, in seconds, that the router will wait before removing the RSA keys that are stored in the eToken after the eToken has been removed from the router.</p> <p><b>Note</b> If this command is not issued, all RSA keys and IPsec tunnels associated with the eToken are torn down immediately after the eToken is removed from the router.</p>
Step 5	<p><b>crypto pki token {token-name   default} max-retries [number]</b></p> <p><b>Example:</b> Router(config)# crypto pki token usbtoken0 max-retries 20</p>	<p>(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the eToken is denied.</p> <ul style="list-style-type: none"> <li>• By default, the value is set at 15.</li> </ul>
Step 6	<p><b>exit</b></p> <p><b>Example:</b> Router(config)# exit</p>	<p>Exits global configuration mode.</p>
Step 7	<p><b>copy usbflash[0-9]:filename destination-url</b></p> <p><b>Example:</b> Router# copy usbflash0:</p>	<p>Copies files from the router to the eToken.</p> <ul style="list-style-type: none"> <li>• <i>destination-url</i>—See the <b>copy</b> command page documentation for a list of supported options.</li> </ul>

	Command or Action	Purpose
Step 8	<code>show usbtoken [0-9] : filename</code>  <b>Example:</b> Router#	(Optional) Displays information about the USB eToken. You can use this command to verify whether the USB eToken has been logged onto the router.
Step 9	<code>crypto pki token token-name logout</code>  <b>Example:</b> Router# <code>crypto pki toke usbtoken0 logout</code>	Logs the router out of the USB eToken.  <b>Note</b> If you want to save any data to the USB eToken, you must log back into the eToken.

## Troubleshooting USB Flash Drives and eTokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB Flash or a USB eToken:

- [The show file systems Command](#)
- [The show usb device Command](#)
- [The show usb controllers Command](#)
- [The dir Command](#)

### The show file systems Command

- Step 1** Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:
- A connection problem with the USB module
  - The Cisco IOS image running on the router does not support a USB module
  - A hardware problem with the USB module itself
- Step 2** Use the **show file systems** command to determine if a USB Flash module is formatted properly. To be compatible with a Cisco router, a USB Flash module must be formatted in a FAT16 format. If that is not the case, the **show file systems** command will display an error indicating an incompatible file system.

Sample output from the **show file systems** command showing a USB Flash module and a USB eToken appear below. The USB module listing appears in the last line of the examples.

```
Router# show file systems
```

```
File Systems:
```

```

      Size (b)      Free (b)      Type  Flags  Prefixes
      -          -          opaque  rw    archive:
      -          -          opaque  rw    system:
      -          -          opaque  rw    null:
      -          -          network rw    tftp:
* 129880064      69414912      disk   rw    flash:#
      491512      486395       nvram  rw    nvram:
      -          -          opaque wo    syslog:
      -          -          opaque rw    xmodem:
      -          -          opaque rw    ymodem:

```

```

-          - network rw rcp:
-          - network rw pram:
-          - network rw ftp:
-          - network rw http:
-          - network rw scp:
-          - network rw https:
-          - opaque ro cns:
63158272  33037312 usbflash rw usbflash0:
32768      858  usbtoken rw  usbtoken1:

```

## The show usb device Command

- Step 1** Use the **show usb device** command to determine if a USB module is supported by Cisco. The sample output for both the USB Flash and the USB eToken that indicates whether or not the module is supported are highlighted in the sample outputs below.

The following sample output is for a USB Flash module:

```

Router# show usb device

Host Controller:1
Address:0x1
Device Configured:YES
Device Supported:YES
Description:DiskOnKey
Manufacturer:M-Sys
Version:2.0
Serial Number:0750D84030316868
Device Handle:0x1000000
USB Version Compliance:2.0
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x8EC
Product ID:0x15
Max. Packet Size of Endpoint Zero:64
Number of Configurations:1
Speed:Full
Selected Configuration:1
Selected Interface:0

Configuration:
  Number:1
  Number of Interfaces:1
  Description:
  Attributes:None
  Max Power:140 mA

Interface:
  Number:0
  Description:
  Class Code:8
  Subclass:6
  Protocol:80
  Number of Endpoints:2

Endpoint:
  Number:1
  Transfer Type:BULK
  Transfer Direction:Device to Host

```

```

Max Packet:64
Interval:0

Endpoint:
  Number:2
  Transfer Type:BULK
  Transfer Direction:Host to Device
  Max Packet:64
  Interval:0

```

The following sample output is for a supported USB eToken:

```

Router# show usb device

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
  Number:1
  Number of Interfaces:1
  Description:
  Attributes:None
  Max Power:60 mA

Interface:
  Number:0
  Description:
  Class Code:255
  Subclass:0
  Protocol:0
  Number of Endpoints:0

```

## The show usb controllers Command

- Step 1** Use the **show usb controllers** command to determine if there is a hardware problem with a USB Flash module. If the **show usb controllers** command displays an error, it indicates a hardware problem in the USB module.

You can also use the **show usb controllers** command to verify that copy operations onto a USB Flash module are occurring successfully. Issuing the **show usb controllers** command after performing a file copy should display successful data transfers.

Sample output for the **show usb controllers** command for a working USB Flash module appears below:

```

Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
  Revision:0x11
  Control:0x80
  Command Status:0x0
  Hardware Interrupt Status:0x24
  Hardware Interrupt Enable:0x80000040
  Hardware Interrupt Disable:0x80000040
  Frame Interval:0x27782EDF
  Frame Remaining:0x13C1
  Frame Number:0xDA4C
  LSThreshold:0x628
  RhDescriptorA:0x19000202
  RhDescriptorB:0x0
  RhStatus:0x0
  RhPort1Status:0x100103
  RhPort2Status:0x100303
  Hardware Configuration:0x3029
  DMA Configuration:0x0
  Transfer Counter:0x1
  Interrupt:0x9
  Interrupt Enable:0x196
  Chip ID:0x3630
  Buffer Status:0x0
  Direct Address Length:0x80A00
  ATL Buffer Size:0x600
  ATL Buffer Port:0x0
  ATL Block Size:0x100
  ATL PTD Skip Map:0xFFFFFFFF
  ATL PTD Last:0x20
  ATL Current Active PTD:0x0
  ATL Threshold Count:0x1
  ATL Threshold Timeout:0xFF

Int Level:1
Transfer Completion Codes:
  Success                :920          CRC                :0
  Bit Stuff              :0           Stall              :0
  No Response            :0           Overrun            :0
  Underrun               :0           Other              :0
  Buffer Overrun         :0           Buffer Underrun    :0

Transfer Errors:
  Canceled Transfers    :2           Control Timeout    :0

Transfer Failures:
  Interrupt Transfer     :0           Bulk Transfer      :0
  Isochronous Transfer  :0           Control Transfer   :0

Transfer Successes:
  Interrupt Transfer     :0           Bulk Transfer      :26
  Isochronous Transfer  :0           Control Transfer   :894

USB Failures:
  Enumeration Failures  :0           No Class Driver Found:0
  Power Budget Exceeded:0

USB MSCD SCSI Class Driver Counters:
  Good Status Failures  :3           Command Fail       :0
  Good Status Timed out:0           Device not Found   :0
  Device Never Opened  :0           Drive Init Fail    :0
  Illegal App Handle    :0           Bad API Command    :0
    
```

```

Invalid Unit Number :0
Application Overflow :0
Control Pipe Stall :0
Device Stalled :0
Device Detached :0
Invalid Logic Unit Num:0
Invalid Argument:0
Device in use :0
Malloc Error :0
Bad Command Code:0
Unknown Error :0

USB Aladdin Token Driver Counters:
Token Inserted :1
Send Insert Msg Fail :0
Dev Entry Add Fail :0
Dev Entry Remove Fail:0
Response Txn Fail :0
Txn Invalid Dev Handle:0
Token Removed :0
Response Txns :434
Request Txns :434
Request Txn Fail:0
Command Txn Fail:0

USB Flash File System Counters:
Flash Disconnected :0
Flash Device Fail :0
Flash startstop Fail :0
Flash Connected :1
Flash Ok :1
Flash FS Fail :0

USB Secure Token File System Counters:
Token Inserted :1
Token FS success :1
Token Max Inserted :0
Token Event :0
Watched Boolean Create Failures:0
Token Detached :0
Token FS Fail :0
Create Talker Failures:0
Destroy Talker Failures:0

```

## The dir Command

- Step 1** Use the **dir** command with the **usbflash[0-9]**: or the **usbtoken[0-9]**: keyword to display all files, directories, and their permission strings on the USB Flash or USB eToken.

The following sample output displays directory information for the USB Flash:

```

Router# dir usbflash0:

Directory of usbflash0:/

   1  -rw-   30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)

```

The following sample output displays directory information for the USB eToken:

```

Router# dir usbtoken1:

Directory of usbtoken1:/

   2  d---         64  Dec 22 2032 05:23:40 +00:00  1000
   5  d---      4096  Dec 22 2032 05:23:40 +00:00  1001
   8  d---         0  Dec 22 2032 05:23:40 +00:00  1002
  10  d---       512  Dec 22 2032 05:23:42 +00:00  1003
  12  d---         0  Dec 22 2032 05:23:42 +00:00  5000
  13  d---         0  Dec 22 2032 05:23:42 +00:00  6000
  14  d---         0  Dec 22 2032 05:23:42 +00:00  7000
  15  ----        940  Jun 27 1992 12:50:42 +00:00  mystartup-config
  16  ----       1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)

```

The following sample output displays directory information for all devices the router is aware of:

```

Router# dir all-filesystems

Directory of archive:/

No files in directory

No space information available
Directory of system:/

   2  drwx           0          <no date>  its
 115  dr-x           0          <no date>  lib
 144  dr-x           0          <no date>  memory
   1  -rw-          1906        <no date>  running-config
 114  dr-x           0          <no date>  vfiles

No space information available
Directory of flash:/

   1  -rw-          30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T

129880064 bytes total (99753984 bytes free)
Directory of nvram:/

 476  -rw-           1947          <no date>  startup-config
 477  ----            46          <no date>  private-config
 478  -rw-           1947          <no date>  underlying-config
   1  -rw-            0          <no date>  ifIndex-table
   2  ----            4          <no date>  rf_cold_starts
   3  ----            14          <no date>  persistent-data

491512 bytes total (486395 bytes free)
Directory of usbflash0:/

   1  -rw-          30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)
Directory of usbtokens1:/

   2  d---            64  Dec 22 2032 05:23:40 +00:00  1000
   5  d---           4096  Dec 22 2032 05:23:40 +00:00  1001
   8  d---            0  Dec 22 2032 05:23:40 +00:00  1002
  10  d---           512  Dec 22 2032 05:23:42 +00:00  1003
  12  d---            0  Dec 22 2032 05:23:42 +00:00  5000
  13  d---            0  Dec 22 2032 05:23:42 +00:00  6000
  14  d---            0  Dec 22 2032 05:23:42 +00:00  7000
  15  ----            940  Jun 27 1992 12:50:42 +00:00  mystartup-config
  16  ----           1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)

```

## Configuration Examples for Secure Token Support

This section contains the following configuration example:

- [Logging Into and Saving RSA Keys to eToken: Example, page 15](#)

## Logging Into and Saving RSA Keys to eToken: Example

The following configuration example shows to how log into the eToken, generate RSA keys, and store the RSA keys onto the eToken:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
    Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
    Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A

% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.

*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
OE30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]

*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **show crypto key mypubkey rsa** command displays stored credentials after they are successfully load from the eToken. Credentials that are stored on the eToken are in the protected area. When storing the credentials on the eToken, the files are stored in a directory called /keystore. However, the key files are hidden from the CLI.

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
Usage:General Purpose Key
```

```

Key is not exportable.
Key Data:
 305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
 732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
 7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
Usage:Encryption Key
Key is not exportable.
Key Data:
 307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
 2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
 56AB8FDC 9911968E DE347FB0 A514A856 B30EAF4 D1F453E1 003CFE65 0CCC6DC7
 21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001

```

## Additional References

The following sections provide references related to USB storage support.

## Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	<i>Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide</i>
eToken and USB Flash data sheet	<i>USB eToken and USB Flash Features Support</i>
File management (loading, copying, and rebooting files)	The section “File Management” in the <i>Cisco IOS Configuration Fundamentals and Network Management Configuration Guide</i> , Release 12.3
Configuring digital certificate encryption	The chapter “Configuring Certification Authority Interoperability” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.3

## Standards

Standards	Title
None	—

## MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

## RFCs

RFCs	Title
None	—

## Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	<a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a>

## Command Reference

This section documents new and modified commands only.

### New Commands

- [crypto pki token change-pin](#)
- [crypto pki token login](#)
- [crypto pki token logout](#)
- [crypto pki token max-retries](#)
- [crypto pki token removal timeout](#)
- [crypto pki token secondary config](#)
- [crypto pki token user-pin](#)
- [debug usb driver](#)
- [show usb driver](#)
- [show usb controllers](#)
- [show usb device](#)
- [show usb driver](#)
- [show usb port](#)
- [show usbtoken](#)
- [show usb tree](#)

### Modified Commands

- [boot config](#)
- [copy](#)
- [delete](#)

- **dir**
- **format**

# boot config

To specify the device and filename of the configuration file from which the router configures itself during initialization (startup), use the **boot config** command in global configuration mode. To remove the specification, use the **no** form of this command.

**boot config** *file-system-prefix*:*[directory/]filename*

**no boot config**

Syntax Description		
<i>file-system-prefix</i> :	File system, followed by a colon (for example, <b>nvr</b> am:, <b>flash</b> :, <b>slot0</b> :, or <b>usbflash[0-9]</b> :, <b>usbtoken[0-9]</b> :).	
<i>directory/</i>	(Optional) File system directory the configuration file is located in, followed by a forward slash (/).	
<i>filename</i>	Name of the configuration file.	

**Defaults** NVRAM (**nvr**am:)

**Command Modes** Global configuration

Command History	Release	Modification
	11.0	This command was introduced.
	12.3(14)T	Support for Class B file system platforms and the following file system prefix options were added: <b>usbflash[0-9]</b> : and <b>usbtoken[0-9]</b> :

**Usage Guidelines** This command is available only on Class A and Class B file system platforms. You set the CONFIG\_FILE environment variable in the current running memory when you use the **boot config** command. This variable specifies the configuration file used for initialization (startup). The configuration file must be an ASCII file located in either NVRAM or Flash memory.



**Note** When you use this global configuration command, you affect only the running configuration. You must save the environment variable setting to your startup configuration to place the information under ROM monitor control and to have the environment variable function as expected. Use the **copy system:running-config nvr**am:startup-config command to save the environment variable from your running configuration to your startup configuration.

The software displays an error message and does not update the CONFIG\_FILE environment variable in the following situations:

- You specify **nvr**am: as the file system, and it contains only a distilled version of the configuration. (A distilled configuration is one that does not contain access lists.)
- You specify a configuration file in the *filename* argument that does not exist or is not valid.

The router uses the NVRAM configuration during initialization when the CONFIG\_FILE environment variable does not exist or when it is null (such as at first-time startup). If the software detects a problem with NVRAM or the configuration it contains, the device enters setup mode.

When you use the **no** form of this command, the router returns to using the default NVRAM configuration file as the startup configuration.

## Examples

In the following example, the first line specifies that the router should use the configuration file named *router-config* located in internal Flash memory to configure itself during initialization. The third line copies the specification to the startup configuration, ensuring that this specification will take effect upon the next reload.

```
Router(config)# boot config flash:router-config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

The following example instructs a Cisco 7500 series router to use the configuration file named *router-config* located on the Flash memory card inserted in the second PCMCIA slot of the RSP card during initialization. The third line copies the specification to the startup configuration, ensuring that this specification will take effect upon the next reload.

```
Router (config)# boot config slot1:router-config
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

## Related Commands

Command	Description
<b>show bootvar</b>	Displays the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, the contents of the BOOTLDR environment variable, and the configuration register setting.

# copy

To copy any file from a source to a destination, use the **copy** command in privileged EXEC mode.

**copy** [**/erase**] [**/verify** | **/noverify**] *source-url destination-url*

Syntax Description	
<b>/erase</b>	(Optional) Erases the destination file system before copying. <b>Note</b> This option is typically provided on platforms with limited memory to allow for an easy way to clear local flash memory space.
<b>/verify</b>	(Optional) Verifies the digital signature of the destination file. If verification fails, the file is deleted from the destination file system. This option applies to Cisco IOS software image files only.
<b>/noverify</b>	(Optional) If the file being copied is an image file, this keyword disables the automatic image verification that occurs after an image is copied. <b>Note</b> This keyword is often issued if the <b>file verify auto</b> command is enabled, which automatically verifies the digital signature of all images that are copied.
<i>source-url</i>	The location URL (or alias) of the source file or directory to be copied. The source can be either local or remote, depending upon whether the file is being downloaded or uploaded.
<i>destination-url</i>	The destination URL (or alias) of the copied file or directory. The destination can be either local or remote, depending upon whether the file is being downloaded or uploaded.

The exact format of the source and destination URLs varies according to the file or directory location. You may enter either an alias keyword for a particular file or a filename that follows the standard Cisco IOS file system syntax (*filesystem:[/filepath][/filename]*).

Table 2 shows two keyword shortcuts to URLs.

**Table 2 Common Keyword Aliases to URLs**

Keyword	Source or Destination
<b>running-config</b>	(Optional) Keyword alias for the <b>system:running-config</b> URL. The <b>system:running-config</b> keyword represents the current running configuration file. This keyword does not work in <b>more</b> and <b>show file</b> EXEC command syntaxes.
<b>startup-config</b>	(Optional) Keyword alias for the <b>nvrn:startup-config</b> URL. The <b>nvrn:startup-config</b> keyword represents the configuration file used during initialization (startup). This file is contained in NVRAM for all platforms except the Cisco 7000 family, which uses the CONFIG_FILE environment variable to specify the startup configuration. The Cisco 4500 series cannot use the <b>copy running-config startup-config</b> command. This keyword does not work in <b>more</b> and <b>show file</b> EXEC command syntaxes.

The following tables list URL prefix keywords by file system type. The available file systems will vary by platform. If you do not specify a URL prefix keyword, the router looks for a file in the current directory.

Table 3 lists URL prefix keywords for Special (opaque) file systems. Table 4 lists them for remote file systems, and Table 5 lists them for local writable storage.

**Table 3 URL Prefix Keywords for Special File Systems**

Keyword	Source or Destination
<b>flh:</b>	Source URL for Flash load helper log files.
<b>modem:</b>	Destination URL for loading modem firmware on to supported networking devices.
<b>null:</b>	Null destination for copies or files. You can copy a remote file to null to determine its size.
<b>nvrasm:</b>	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
<b>system:</b>	Source or destination URL for system memory, which includes the running configuration.
<b>xmodem:</b>	Source or destination for a file from a network machine that uses the Xmodem protocol.
<b>ymodem:</b>	Source or destination for a file from a network machine that uses the Ymodem protocol.

**Table 4 URL Prefix Keywords for Remote File Systems**

Keyword	Source or Destination
<b>ftp:</b>	Source or destination URL for File Transfer Protocol (FTP) network server. The syntax for this alias is as follows: <b>ftp:</b> [[[/username[:password]@]location]/directory]/filename.
<b>http://</b>	Source or destination URL for a Hypertext Transfer Protocol (HTTP) server (also called a web server). The syntax for this alias is as follows: <b>http://</b> [[username:password]@]{hostname   host-ip}[/filepath]/filename
<b>https://</b>	Source or destination URL for a Secure HTTP (HTTPS) server. HTTPS uses Secure Socket Layer (SSL) encryption. The syntax for this alias is as follows: <b>https://</b> [[username:password]@]{hostname   host-ip}[/filepath]/filename
<b>rcp:</b>	Source or destination URL for a remote copy protocol (rcp) network server. The syntax for this alias is as follows: <b>rcp:</b> [[[/username@]location]/directory]/filename
<b>scp:</b>	Source or destination URL for a network server that supports Secure Shell (SSH) and accepts copies of files using the secure copy protocol (scp). The syntax for this alias is as follows: <b>scp:</b> //username@location[/directory]/filename]
<b>tftp:</b>	Source or destination URL for a TFTP network server. The syntax for this alias is as follows: <b>tftp:</b> [[/location]/directory]/filename.

**Table 5 URL Prefix Keywords for Local Writable Storage File Systems**

Alias	Source or Destination
<b>bootflash:</b>	Source or destination URL for boot Flash memory.
<b>disk0: and disk1:</b>	Source or destination URL of disk-based media.
<b>flash:</b>	Source or destination URL for flash memory. This alias is available on all platforms. For platforms that lack a flash: device, note that <b>flash:</b> is aliased to <b>slot0:</b> , allowing you to refer to the main Flash memory storage area on all platforms.
<b>slavebootflash:</b>	Source or destination URL for internal Flash memory on the slave RSP card of a router configured for HSA.
<b>slaveram:</b>	NVRAM on a slave RSP card of a router configured for HSA.
<b>slaveslot0:</b>	Source or destination URL of the first PCMCIA card on a slave RSP card of a router configured for HSA.
<b>slaveslot1:</b>	Source or destination URL of the second PCMCIA slot on a slave RSP card of a router configured for HSA.
<b>slot0:</b>	Source or destination URL of the first PCMCIA Flash memory card.
<b>slot1:</b>	Source or destination URL of the second PCMCIA Flash memory card.
<b>usbflash[0-9]:</b>	Source or destination URL for the USB flash drive that has been plugged into the router.
<b>usbtoken[0-9]:</b>	Source or destination URL for the USB eToken that has been plugged into the router.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	11.3 T	This command was introduced.
	12.3(2)T	<ul style="list-style-type: none"> <li>The <b>http://</b> and <b>https://</b> keywords were added as supported remote source locations (file system URL prefixes) for files.</li> <li>This command was enhanced to support copying files to servers that support Secure Shell (SSH) and the secure copy protocol (scp).</li> </ul>
	12.2(18)S	The <b>/verify</b> and <b>/noverify</b> keywords were added.
	12.0(26)S	The <b>/verify</b> and <b>/noverify</b> keywords were integrated into Cisco IOS Release 12.0(26)S.
	12.3(4)T	The <b>/verify</b> and <b>/noverify</b> keywords were integrated into Cisco IOS Release 12.3(4)T.
	12.3(7)T	The <b>http://</b> and <b>https://</b> keywords were enhanced to support file uploads.
	12.3(14)T	The <b>usbflash[0-9]:</b> and <b>usbtoken[0-9]:</b> keywords were added to support USB storage.

## Usage Guidelines

The fundamental function of the copy command is to allow you to copy a file (such as a system image or configuration file) from one location to another location. The source and destination for the file is specified using a Cisco IOS File System URL, which allows you to specify any supported local or remote file location. The file system being used (such as a local memory source, or a remote server) dictates the syntax used in the command.

You can enter on the command line all necessary source- and destination-URL information and the username and password to use, or you can enter the **copy** command and have the router prompt you for any missing information.

For local file systems, two commonly used aliases exist for the **system:running-config** and **nvram:startup-config** files; these aliases are **running-config** and **startup-config**, respectively.



### Timesaver

Aliases are used to cut down on the amount of typing you need to perform. For example, it is easier to type **copy run start** (the abbreviated form of the **copy running-config startup-config** command) than it is to type **copy system:r nvram:s** (the abbreviated form of the **copy system:running-config nvram:startup-config** command). These aliases also allow you to continue using some of the common commands used in previous versions of Cisco IOS software.

The entire copying process may take several minutes and differs from protocol to protocol and from network to network.

The colon is required after the file system URL prefix keywords (such as **flash**). In some cases, file system prefixes that did not require colons in earlier software releases are allowed for backwards compatibility, but use of the colon is recommended.

In the URL syntax for **ftp:**, **http:**, **https:**, **rcp:**, **scp:** and **tftp:**, the location is either an IP address or a host name. The filename is specified relative to the directory used for file transfers.

The following sections contain usage guidelines for the following topics:

- [Understanding Invalid Combinations of Source and Destination, page 25](#)
- [Understanding Character Descriptions, page 25](#)
- [Understanding Partitions, page 25](#)
- [Using rcp, page 25](#)
- [Using FTP, page 26](#)
- [Using HTTP\(S\), page 26](#)
- [Storing Images on Servers, page 27](#)
- [Copying from a Server to Flash Memory, page 27](#)
- [Verifying Images, page 27](#)
- [Copying a Configuration File from a Server to the Running Configuration, page 28](#)
- [Copying a Configuration File from a Server to the Startup Configuration, page 28](#)
- [Storing the Running or Startup Configuration on a Server, page 28](#)
- [Saving the Running Configuration to the Startup Configuration, page 28](#)
- [Using CONFIG\\_FILE, BOOT, and BOOTLDR Environment Variables, page 29](#)
- [Using the Copy Command with the Dual RSP Feature, page 29](#)

### Understanding Invalid Combinations of Source and Destination

Some invalid combinations of source and destination exist. Specifically, you cannot copy:

- From a running configuration to a running configuration
- From a startup configuration to a startup configuration
- From a device to the same device (for example, the **copy flash: flash:** command is invalid)

### Understanding Character Descriptions

Table 6 describes the characters that you may see during processing of the **copy** command.

**Table 6** *copy Character Descriptions*

Character	Description
!	For network transfers, an exclamation point indicates that the copy process is taking place. Each exclamation point indicates the successful transfer of ten packets (512 bytes each).
.	For network transfers, a period indicates that the copy process timed out. Many periods in a row typically mean that the copy process may fail.
O	For network transfers, an uppercase O indicates that a packet was received out of order and the copy process may fail.
e	For Flash erasures, a lowercase e indicates that a device is being erased.
E	An uppercase E indicates an error. The copy process may fail.
V	A series of uppercase Vs indicates the progress during the verification of the image checksum.

### Understanding Partitions

You cannot copy an image or configuration file to a Flash partition from which you are currently running. For example, if partition 1 is running the current system image, copy the configuration file or image to partition 2. Otherwise, the copy operation will fail.

You can identify the available Flash partitions by entering the **show file system EXEC** command.

### Using rcp

The rcp requires a client to send a remote username upon each rcp request to a server. When you copy a configuration file or image between the router and a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The remote username specified in the **copy** command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to process, an account must be defined on the network server for the remote username. If the network administrator of the destination server did not establish an account for the remote username, this command will not run. If the server has a directory structure, the configuration

file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that username as the remote username.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, add an entry to the `.rhosts` file for the remote user on the rcp server. Suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router IP address translates to `Router1.company.com`, then the `.rhosts` file for `User0` on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more details.

If you are using a personal computer as a file server, the computer must support the remote shell protocol (rsh).

### Using FTP

The FTP protocol requires a client to send a username and password with each FTP request to a remote FTP server. Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a default username and password for all copy operations to or from an FTP server. Include the username in the **copy** command syntax if you want to specify a username for that copy operation only.

When you copy a file from the router to a server using FTP, the Cisco IOS software sends the first valid username that it encounters in the following sequence:

1. The username specified in the **copy** command, if a username is specified.
2. The username set by the **ip ftp username** command, if the command is configured.
3. Anonymous.

The router sends the first valid password in the following list:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password `username@routername.domain`. The variable `username` is the username associated with the current session, `routername` is the configured host name, and `domain` is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that username as the remote username.

Refer to the documentation for your FTP server for details on setting up the server.

### Using HTTP(S)

Copying a file to or from a remote HTTP or HTTPS server, to or from a local file system, is performed using the embedded Secure HTTP client that is integrated in Cisco IOS software. The HTTP client is enabled by default.

Downloading files from a remote HTTP or HTTPS server is performed using the HTTP client integrated in Cisco IOS software.

If a username and password are not specified in the **copy** command syntax, the system uses the default HTTP client username and password, if configured.

When you copy a file from a remote HTTP(S) server, the Cisco IOS software sends the first valid username that it encounters in the following sequence:

1. The username specified in the **copy** command, if a username is specified.
2. The username set by the **ip http client username** command, if the command is configured.
3. Anonymous.

The router sends the first valid password in the following list:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip http client password** command, if the command is configured.
3. The router forms the password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

### Storing Images on Servers

Use the **copy flash: destination-url** command (for example, **copy flash: tftp:**) to copy a system image or boot image from Flash memory to a network server. You can use the copy of the image as a backup copy. Also, you can also use the image backup file to verify that the image in Flash memory is the same as that in the original file.

### Copying from a Server to Flash Memory

Use the **copy destination-url flash:** command (for example, **copy tftp: flash:**) to copy an image from a server to Flash memory.

On Class B file system platforms, the system provides an option to erase existing Flash memory before writing onto it.



#### Note

---

Verify the image in Flash memory before booting the image.

---

### Verifying Images

When copying a new image to your router, you should confirm that the image was not corrupted during the copy process. You can verify the integrity of the image in any of the following ways:

- Depending on the destination file system type, a checksum for the image file may be displayed when the **copy** command completes. You can verify this checksum by comparing it to the checksum value provided for your image file on Cisco.com.



#### Caution

---

If the checksum values do not match, do not reboot the router. Instead, reissue the **copy** command and compare the checksums again. If the checksum is repeatedly wrong, copy the original image back into Flash memory *before* you reboot the router from Flash memory. If you have a corrupted image in Flash memory and try to boot from Flash memory, the router will start the system image contained in ROM (assuming booting from a network server is not configured). If ROM does not contain a fully functional system image, the router might not function and will need to be reconfigured through a direct console port connection.

---

- Use the **/verify** keyword.

- Enable automatic image verification by default by issuing the **file verify auto** command. This command will automatically check the integrity of each file that is copied via the **copy** command (without specifying the **/verify** option) to the router unless the **/noverify** keyword is specified.
- Use the UNIX 'diff' command. This method can also be applied to file types other than Cisco IOS images. If you suspect that a file is corrupted, copy the suspect file and the original file to a UNIX server. (The file names may need to be modified if you try to save the files in the same directory.) Then run the UNIX 'diff' command on the two files. If there is no difference, then the file has not been corrupted.

#### Copying a Configuration File from a Server to the Running Configuration

Use the **copy {ftp: | rcp: | scp: | tftp:} running-config** command to load a configuration file from a network server to the running configuration of the router. (Note that **running-config** is the alias for the **system:running-config** keyword.) The configuration will be added to the running configuration as if the commands were typed in the command-line interface (CLI). Thus, the resulting configuration file will be a combination of the previous running configuration and the loaded configuration file, with the loaded configuration file having precedence.

You can copy either a host configuration file or a network configuration file. Accept the default value of *host* to copy and load a host configuration file containing commands that apply to one network server in particular. Enter *network* to copy and load a network configuration file containing commands that apply to all network servers on a network.

#### Copying a Configuration File from a Server to the Startup Configuration

Use the **copy {ftp: | rcp: | scp: | tftp:} nvram:startup-config** command to copy a configuration file from a network server to the router startup configuration. These commands replace the startup configuration file with the copied configuration file.

#### Storing the Running or Startup Configuration on a Server

Use the **copy system:running-config {ftp: | rcp: | scp: | tftp:}** command to copy the current configuration file to a network server using FTP, rcp, scp, or TFTP. Use the **copy nvram:startup-config {ftp: | rcp: | scp: | tftp:}** command to copy the startup configuration file to a network server.

The configuration file copy can serve as a backup copy.

#### Saving the Running Configuration to the Startup Configuration

Use the **copy system:running-config nvram:startup-config** command to copy the running configuration to the startup configuration.



#### Note

---

Some specific commands might not get saved to NVRAM. You will need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a listing of these settings so you can quickly reconfigure your router after rebooting.

---

If you issue the **copy system:running-config nvram:startup-config** command from a bootstrap system image, a warning will instruct you to indicate whether you want your previous NVRAM configuration to be overwritten and configuration commands to be lost. This warning does not appear if NVRAM contains an invalid configuration or if the previous configuration in NVRAM was generated by a bootstrap system image.

On all platforms except Class A file system platforms, the **copy system:running-config nvram:startup-config** command copies the currently running configuration to NVRAM.

On the Class A Flash file system platforms, the **copy system:running-config nvram:startup-config** command copies the currently running configuration to the location specified by the CONFIG\_FILE environment variable. This variable specifies the device and configuration file used for initialization. When the CONFIG\_FILE environment variable points to NVRAM or when this variable does not exist (such as at first-time startup), the software writes the current configuration to NVRAM. If the current configuration is too large for NVRAM, the software displays a message and stops executing the command.

When the CONFIG\_FILE environment variable specifies a valid device other than **nvram:** (that is, **flash:**, **bootflash:**, **slot0:**, or **slot1:**), the software writes the current configuration to the specified device and filename, and stores a distilled version of the configuration in NVRAM. A distilled version is one that does not contain access list information. If NVRAM already contains a copy of a complete configuration, the router prompts you to confirm the copy.

### Using CONFIG\_FILE, BOOT, and BOOTLDR Environment Variables

For the Class A Flash file system platforms, specifications are as follows:

- The CONFIG\_FILE environment variable specifies the configuration file used during router initialization.
- The BOOT environment variable specifies a list of bootable images on various devices.
- The BOOTLDR environment variable specifies the Flash device and filename containing the rxboot image that ROM uses for booting.
- Cisco 3600 routers do not use a dedicated boot helper image (rxboot), which many other routers use to help with the boot process. Instead, the BOOTLDR ROM monitor environment variable identifies the Flash memory device and filename that are used as the boot helper; the default is the first system image in Flash memory.

To view the contents of environment variables, use the **show bootvar EXEC** command. To modify the CONFIG\_FILE environment variable, use the **boot config** global configuration command. To modify the BOOTLDR environment variable, use the **boot bootldr** global configuration command. To modify the BOOT environment variable, use the **boot system** global configuration command. To save your modifications, use the **copy system:running-config nvram:startup-config** command.

When the destination of a **copy** command is specified by the CONFIG\_FILE or BOOTLDR environment variable, the router prompts you for confirmation before proceeding with the copy. When the destination is the only valid image in the BOOT environment variable, the router also prompts you for confirmation before proceeding with the copy.

### Using the Copy Command with the Dual RSP Feature

The Dual RSP feature allows you to install two Route Switch Processor (RSP) cards in a single router on the Cisco 7507 and Cisco 7513 platforms.

On a Cisco 7507 or Cisco 7513 router configured for Dual RSPs, if you copy a file to **nvram:startup-configuration** with automatic synchronization disabled, the system prompts whether you also want to copy the file to the slave startup configuration. The default answer is **yes**. If automatic synchronization is enabled, the system automatically copies the file to the slave startup configuration each time you use a **copy** command with **nvram:startup-configuration** as the destination.

### Examples

The following examples illustrate uses of the **copy** command:

- [Verifying the Integrity of the Image Before It Is Copied Example, page 30](#)
- [Copying an Image from a Server to Flash Memory Examples, page 30](#)

- [Saving a Copy of an Image on a Server Examples, page 32](#)
- [Copying a Configuration File from a Server to the Running Configuration Example, page 34](#)
- [Copying a Configuration File from a Server to the Startup Configuration Example, page 34](#)
- [Copying the Running Configuration to a Server Example, page 35](#)
- [Copying the Startup Configuration to a Server Example, page 35](#)
- [Saving the Current Running Configuration Example, page 35](#)
- [Moving Configuration Files to Other Locations Examples, page 35](#)
- [Copying a File from a Remote Web Server Examples, page 37](#)
- [Copying an Image from the Master RSP Card to the Slave RSP Card Example, page 37](#)

**Verifying the Integrity of the Image Before It Is Copied Example**

The following example shows how to specify image verification before copying an image:

```
Router# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:

Destination filename [c7200-js-mz]?
Accessing tftp://10.1.1.1/jdoe/c7200-js-mz...
Loading jdoe/c7200-js-mz from 10.1.1.1 (via FastEthernet0/0):!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 19879944 bytes]

19879944 bytes copied in 108.632 secs (183003 bytes/sec)
Verifying file integrity of disk0:/c7200-js-mz
.....
.....
.....Done!
Embedded Hash MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash MD5 :44A7B9BDDD9638128C35528466318183

Signature Verified
```

**Copying an Image from a Server to Flash Memory Examples**

The following examples use a **copy rcp:**, **copy tftp:**, or **copy ftp:** command to copy an image file from a server to Flash memory:

- [Copying an Image from a Server to Flash Memory Example, page 30](#)
- [Copying an Image from a Server to a Flash Memory Using Flash Load Helper Example, page 31](#)
- [Copying an Image from a Server to a Flash Memory Card Partition Example, page 32](#)

**Copying an Image from a Server to Flash Memory Example**

The following example copies a system image named file1 from the remote rcp server with an IP address of 172.16.101.101 to Flash memory. On Class B file system platforms, the Cisco IOS software allows you to first erase the contents of Flash memory to ensure that enough Flash memory is available to accommodate the system image.

```
Router# copy rcp://netadmin@172.16.101.101/file1 flash:file1

Destination file name [file1]?
Accessing file 'file1' on 172.16.101.101...
Loading file1 from 172.16.101.101 (via Ethernet0): ! [OK]
```



### Copying an Image from a Server to a Flash Memory Card Partition Example

The following example copies the file c3600-i-mz from the rcp server at IP address 172.23.1.129 to the Flash memory card in slot 0 of a Cisco 3600 series router, which has only one partition. As the operation progresses, the Cisco IOS software prompts you to erase the files on the Flash memory PC card to accommodate the incoming file. This entire operation takes 18 seconds to perform, as indicated at the end of the example.

```
Router# copy rcp: slot0:

PCMCIA Slot0 flash

Partition  Size  Used   Free   Bank-Size  State      Copy Mode
-----  -
1          4096K  3068K  1027K   4096K     Read/Write Direct
2          4096K  1671K  2424K   4096K     Read/Write Direct
3          4096K   0K    4095K   4096K     Read/Write Direct
4          4096K  3825K  270K    4096K     Read/Write Direct

[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]

PCMCIA Slot0 flash directory, partition 1:
File Length  Name/status
  1  3142288  c3600-j-mz.test
[3142352 bytes used, 1051952 available, 4194304 total]
Address or name of remote host [172.23.1.129]?
Source file name? /tftpboot/images/c3600-i-mz
Destination file name [/tftpboot/images/c3600-i-mz]?
Accessing file '/tftpboot/images/c3600-i-mz' on 172.23.1.129...
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz: ! [OK]

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy '/tftpboot/images/c3600-i-mz' from server
  as '/tftpboot/images/c3600-i-mz' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Verifying checksum... OK (0xF89A)
Flash device copy took 00:00:18 [hh:mm:ss]
```

### Saving a Copy of an Image on a Server Examples

The following examples use **copy** commands to copy image files to a server for storage:

- [Copy an Image from Flash Memory to an rcp Server Example, page 33](#)
- [Copy an Image from Flash Memory to an SSH Server Using scp Example, page 33](#)
- [Copy an Image from a Partition of Flash Memory to a Server Example, page 33](#)
- [Copying an Image from a Flash Memory File System to an FTP Server Example, page 34](#)
- [Copying an Image from Boot Flash Memory to a TFTP Server Example, page 34](#)

### Copy an Image from Flash Memory to an rcp Server Example

The following example copies a system image from Flash Memory to an rcp server using the default remote username. Because the rcp server address and filename are not included in the command, the router prompts for it.

```
Router# copy flash: rcp:

IP address of remote host [255.255.255.255]? 172.16.13.110
Name of file to copy? gsxx
writing gsxx - copy complete
```

### Copy an Image from Flash Memory to an SSH Server Using scp Example

The following example shows how to use scp to copy a system image from Flash Memory to a server that supports SSH:

```
Router# copy flash:c4500-ik2s-mz.scp scp://user1@host1/
Address or name of remote host [host1]?
Destination username [user1]?
Destination filename [c4500-ik2s-mz.scp]?
Writing c4500-ik2s-mz.scp
Password:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Before you can use the server-side functionality, SSH, authentication, and authorization must be properly configured so the router can determine whether a user is at the right privilege level. The scp server-side functionality is configured with the **ip scp server enable** command.

### Copy an Image from a Partition of Flash Memory to a Server Example

The following example copies an image from a particular partition of Flash memory to an rcp server using a remote username of netadmin1.

The system will prompt if there are two or more partitions. If the partition entered is not valid, the process terminates. You have the option to enter a partition number, a question mark (?) for a directory display of all partitions, or a question mark and a number (?number) for a directory display of a particular partition. The default is the first partition.

```
Router# configure terminal
Router# ip rcmd remote-username netadmin1
Router# end
Router# copy flash: rcp:
System flash partition information:
Partition  Size      Used      Free      Bank-Size  State      Copy-Mode
   1         4096K    2048K    2048K    2048K      Read Only  RXBOOT-FLH
   2         4096K    2048K    2048K    2048K      Read/Write Direct

[Type ?<number> for partition directory; ? for full directory; q to abort]
Which partition? [1] 2

System flash directory, partition 2:
File Length Name/status
  1  3459720 master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [ABC.CISCO.COM]?
Source file name? master/igs-bfpx.100-4.3
Destination file name [master/igs-bfpx.100-4.3]?
Verifying checksum for 'master/igs-bfpx.100-4.3' (file # 1)... OK
Copy 'master/igs-bfpx.100-4.3' from Flash to server
as 'master/igs-bfpx.100-4.3'? [yes/no] yes
!!!!...
Upload to server done
Flash copy took 0:00:00 [hh:mm:ss]
```

### Copying an Image from a Flash Memory File System to an FTP Server Example

The following example copies the file c3600-i-mz from partition 1 of the Flash memory card in slot 0 to an FTP server at IP address 172.23.1.129:

```
Router# show slot0: partition 1

PCMCIA Slot0 flash directory, partition 1:
File Length Name/status
  1 1711088 c3600-i-mz
[1711152 bytes used, 2483152 available, 4194304 total]

Router# copy slot0:1:c3600-i-mz ftp://myuser:mypass@172.23.1.129/c3600-i-mz
Verifying checksum for '/tftpboot/cisco_rules/c3600-i-mz' (file # 1)... OK
Copy '/tftpboot/cisco_rules/c3600-i-mz' from Flash to server
  as 'c3700-i-mz'? [yes/no] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:23 [hh:mm:ss]
```

### Copying an Image from Boot Flash Memory to a TFTP Server Example

The following example copies an image from boot Flash memory to a TFTP server:

```
Router# copy bootflash:file1 tftp://192.168.117.23/file1

Verifying checksum for 'file1' (file # 1)... OK
Copy 'file1' from Flash to server
  as 'file1'? [yes/no] y
!!!!...
Upload to server done
Flash copy took 0:00:00 [hh:mm:ss]
```

### Copying a Configuration File from a Server to the Running Configuration Example

The following example copies and runs a configuration filename host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101:

```
Router# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config

Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

### Copying a Configuration File from a Server to the Startup Configuration Example

The following example copies a configuration file host2-confg from a remote FTP server to the startup configuration. The IP address is 172.16.101.101, the remote username is netadmin1, and the remote password is ftppass.

```
Router# copy ftp://netadmin1:ftppass@172.16.101.101/host2-confg nvram:startup-config
Configure using rtr2-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file rtr2-confg:![OK]
[OK]
Router#
%SYS-5-CONFIG_NV:Non-volatile store configured from rtr2-config by
FTP from 172.16.101.101
```

### Copying the Running Configuration to a Server Example

The following example specifies a remote username of netadmin1. Then it copies the running configuration file named rtr2-config to the netadmin1 directory on the remote host with an IP address of 172.16.101.101.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Router(config)# end
Router# copy system:running-config rcp:
Remote host[]? 172.16.101.101

Name of configuration file to write [Rtr2-config]?
Write file rtr2-config on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
```

### Copying the Startup Configuration to a Server Example

The following example copies the startup configuration to a TFTP server:

```
Router# copy nvram:startup-config tftp:

Remote host[]? 172.16.101.101

Name of configuration file to write [rtr2-config]? <cr>
Write file rtr2-config on host 172.16.101.101?[confirm] <cr>
![OK]
```

### Saving the Current Running Configuration Example

The following example copies the running configuration to the startup configuration. On a Class A Flash file system platform, this command copies the running configuration to the startup configuration specified by the CONFIG\_FILE variable.

```
copy system:running-config nvram:startup-config
```

The following example shows the warning that the system provides if you try to save configuration information from bootstrap into the system:

```
Router(boot)# copy system:running-config nvram:startup-config

Warning: Attempting to overwrite an NVRAM configuration written
by a full system image. This bootstrap software does not support
the full configuration command set. If you perform this command now,
some configuration commands may be lost.
Overwrite the previous NVRAM configuration?[confirm]
```

Enter **no** to escape writing the configuration information to memory.

### Moving Configuration Files to Other Locations Examples

On some routers, you can store copies of configuration files on a Flash memory device. Five examples follow:

- [Copying the Startup Configuration to a Flash Memory Device Example, page 36](#)
- [Copying the Running Configuration to a Flash Memory Device Example, page 36](#)
- [Copying to the Running Configuration from a Flash Memory Device Example, page 36](#)
- [Copying to the Startup Configuration from a Flash Memory Device Example, page 36](#)
- [Copying a Configuration File from one Flash Device to Another Example, page 36](#)

### Copying the Startup Configuration to a Flash Memory Device Example

The following example copies the startup configuration file (specified by the CONFIG\_FILE environment variable) to a Flash memory card inserted in slot 0:

```
copy nvram:startup-config slot0:router-config
```

### Copying the Running Configuration to a Flash Memory Device Example

The following example copies the running configuration from the router to the Flash memory PC card in slot 0:

```
Router# copy system:running-config slot0:berlin-cfg
```

```
Building configuration...
```

```
5267 bytes copied in 0.720 secs
```

### Copying to the Running Configuration from a Flash Memory Device Example

The following example copies the file named ios-upgrade-1 from the Flash memory card in slot 0 to the running configuration:

```
Router# copy slot0:4:ios-upgrade-1 system:running-config
```

```
Copy 'ios-upgrade-1' from flash device
  as 'running-config' ? [yes/no] yes
```

### Copying to the Startup Configuration from a Flash Memory Device Example

The following example copies the router-image file from the Flash memory to the startup configuration:

```
copy flash:router-image nvram:startup-config
```

### Copying a Configuration File from one Flash Device to Another Example

The following example copies the file running-config from the first partition in internal Flash memory to the Flash memory PC card in slot 1. The checksum of the file is verified, and its copying time of 30 seconds is displayed.

```
Router# copy flash: slot1:
```

```
System flash
```

Partition	Size	Used	Free	Bank-Size	State	Copy Mode
1	4096K	3070K	1025K	4096K	Read/Write	Direct
2	16384K	1671K	14712K	8192K	Read/Write	Direct

```
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
```

```
System flash directory, partition 1:
```

```
File Length Name/status
  1 3142748 dirt/images/mars-test/c3600-j-mz.latest
  2 850 running-config
[3143728 bytes used, 1050576 available, 4194304 total]
```

```
PCMCIA Slot1 flash directory:
```

```
File Length Name/status
  1 1711088 dirt/images/c3600-i-mz
  2 850 running-config
[1712068 bytes used, 2482236 available, 4194304 total]
```

```
Source file name? running-config
```

```
Destination file name [running-config]?
```

```

Verifying checksum for 'running-config' (file # 2)... OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy 'running-config' from flash: device
  as 'running-config' into slot1: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
!
[OK - 850/4194304 bytes]

Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum... OK (0x16)

```

### Copying a File from a Remote Web Server Examples

In the following example, the file config1 is copied from a remote server to Flash memory using HTTP:

```
Router# copy http://www.example.com:8080/configs/config1 flash:config1
```

In the following example, a default username and password for HTTP Client communications is configured, and then the file sample.scr is copied from a secure HTTP server using HTTPS:

```

Router# configure terminal
Router(config)# ip http client username joeuser
Router(config)# ip http client password letmein
Router(config)# end
Router# copy https://www.example_secure.com/scripts/sample.scr flash:

```

In the following example, an HTTP proxy server is specified before using the **copy http://** command:

```

Router# configure terminal
Router(config)# ip http client proxy-server edge2 proxy-port 29
Router(config)# end
Router# copy http://www.example.com/configs/config3 flash:/configs/config3

```

### Copying an Image from the Master RSP Card to the Slave RSP Card Example

The following example copies the router-image file from the Flash memory card inserted in slot 1 of the master RSP card to slot 0 of the slave RSP card in the same router:

```
Router# copy slot1:router-image slaveslot0:
```

#### Related Commands

Command	Description
<b>boot config</b>	Specifies the device and filename of the configuration file from which the router configures itself during initialization (startup).
<b>boot system</b>	Specifies the system image that the router loads at startup.
<b>cd</b>	Changes the default directory or file system.
<b>copy xmodem: flash:</b>	Copies any file from a source to a destination.
<b>copy ymodem: flash:</b>	Copies any file from a source to a destination.
<b>delete</b>	Deletes a file on a Flash memory device.
<b>dir</b>	Displays a list of files on a file system.
<b>erase</b>	Erases a file system.
<b>ip rcmd remote-username</b>	Configures the remote username to be used when requesting a remote copy using rcp.
<b>ip scp server enable</b>	Enables scp server-side functionality.

Command	Description
<b>reload</b>	Reloads the operating system.
<b>show bootvar</b>	Displays the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, the contents of the BOOTLDR environment variable, and the configuration register setting.
<b>show (Flash file system)</b>	Displays the layout and contents of a Flash memory file system.
<b>slave auto-sync config</b>	Turns on automatic synchronization of configuration files for a Cisco 7507 or Cisco 7513 router that is configured for Dual RSP Backup.
<b>verify bootflash:</b>	Either of the identical <b>verify bootflash:</b> or <b>verify bootflash</b> commands replaces the <b>copy verify bootflash</b> command. Refer to the <b>verify</b> command for more information.

# crypto pki token change-pin

To change the user PIN on the USB eToken, use the **crypto pki token change-pin** command in privileged EXEC mode.

```
crypto pki token token-name [admin] change-pin [pin]
```

Syntax Description	<i>token-name</i>	Name of USB token specified via the <b>crypto pki token login</b> command.
	<b>admin</b>	(Optional) The router will change the administrative PIN on the USB token. If this keyword is not issued, the router will change the user pin.
	<i>pin</i>	(Optional) User PIN required to access the etoken.

**Command Default** None

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines**

If you want to change the administrative PIN on the token, you must be logged into the eToken as an admin via the **crypto pki token admin login** command.

After the user PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.

**Examples** The following example shows that the user PIN was changed to 1234:

```
crypto pki token usbtokens0 admin login 5678
crypto pki token usbtokens0 change-pin 1234
```

Related Commands	Command	Description
	<b>crypto pki token login</b>	Logs into the USB eToken.
	<b>crypto pki token max-retries</b>	Sets the maximum number of allowed failed login attempts.

# crypto pki token login

To log into the USB eToken, use the **crypto pki token login** command in privileged EXEC mode.

**crypto pki token** *token-name* [**admin**] **login** [*pin*]

Syntax Description		
	<i>token-name</i>	Name of USB eToken.
	<b>admin</b>	(Optional) The router will attempt to log into the token as an administrator. If this keyword is not issued, the router will attempt to log into the token as a user.  <b>Note</b> If you want to change the PIN via the <b>crypto pki token change-pin</b> command, you must issue this keyword.
	<i>pin</i>	(Optional) User PIN required to access the token. If a user PIN is not specified, the default PIN, 1234567890, is used.

**Command Default** None

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** This command allows you to manually log into a USB eToken. To automatically log into an eToken, issue the **crypto pki token user-pin** command, which allows you to create a PIN for automatic login.

**Examples** The following example shows how to log into the USB eToken manually:

```
crypto pki token usbtoken0:login 1234567890
```

Related Commands	Command	Description
	<b>crypto pki token logout</b>	Logs the router out of the USB eToken.

# crypto pki token logout

To log the router out of the USB eToken, use the **crypto pki token logout** command in privileged EXEC mode.

**crypto pki token** *token-name* **logout**

<b>Syntax Description</b>	<i>token-name</i>	Name of USB eToken specified via the <b>crypto pki token login</b> command.
---------------------------	-------------------	---

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.3(14)T	This command was introduced.

<b>Usage Guidelines</b>	If you want to save any data to the USB eToken, you must log back into the eToken.
-------------------------	--

<b>Examples</b>	<p>The following example shows how to successfully log out of a USB eToken:</p> <pre>crypto pki token usbtoken0:logout Token eToken is usbtoken0  Token logout from usbtoken0 (eToken) successful *Jan 28 05:46:59.544:%CRYPTO-6-TOKENLOGOUT:Cryptographic Token eToken Logout Successful</pre>
-----------------	---

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>crypto pki token login</b>	Logs into the USB eToken.

# crypto pki token max-retries

To set the maximum number of allowed failed login attempts, use the **crypto pki token max-retries** command in global configuration mode. To return to the default functionality (which is 15 failed login attempts), use the **no** form of this command.

**crypto pki token** {*token-name* | **default**} **max-retries** [*number*]

**no crypto pki token** {*token-name* | **default**} **max-retries** [*number*]

<b>Syntax Description</b>	<i>token-name</i>	Name of USB token that the router will log into.
	<b>default</b>	Default value is to be used.
	<i>number</i>	(Optional) Number of consecutive failed login attempts the router will allow before locking out the user. Available range: 0 to 15. Default value is 15.

**Defaults** 15 failed login attempts are allowed

**Command Modes** Global configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.3(14)T	This command was introduced.

**Usage Guidelines** After the user PIN is changed via the **crypto pki token change-pin command**, the login failure count is automatically reset to 15; however, it is recommended that the login failure count be set to zero.

**Examples** The following example shows how to change the allowed maximum number of failed login attempts to 20:

```
crypto pki token usbtok0 max-retries 20
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>crypto pki token change-pin</b>	Changes the user PIN number on the USB eToken.
	<b>crypto pki token login</b>	Logs into the USB eToken.

# crypto pki token removal timeout

To set the time interval that the router waits before removing the Rivest, Shamir, and Adelman (RSA) keys that are stored in the eToken, use the **crypto pki token removal timeout** command in global configuration mode. To return to the default functionality (which is no timeout), use the **no** form of this command.

**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]

**no crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]

Syntax Description	
<i>token-name</i>	Name of USB eToken that is being removed from the router.
<b>default</b>	Default value, which is automatic RSA key removal, is to be used.
<i>seconds</i>	(Optional) Time interval, in seconds, that the router waits before removing the RSA keys and tearing down IP Security (IPSec) tunnels associated with the specified eToken. Available range: 0 to 480.
<b>Note</b>	If a time interval is not specified, all RSA keys and associated tunnels are immediately torn down after the eToken is removed from the router.

**Defaults** RSA keys are automatically removed after the eToken is removed from the router.

**Command Modes** Global configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** After the eToken is removed from the router, you can clear from your router any RSA keys that were obtained from the eToken; all IPSec tunnels that used those RSA keys for authentication are also torn down. Both the keys and tunnels are immediately cleared unless otherwise specified via the **crypto pki token removal timeout** command.

Although the RSA keys remain on the eToken, they can only be accessed with the correct PIN. Too many unsuccessful attempts to log into the eToken will disable the PIN and any further login attempts will be refused.



**Note**

The **no** version of this command does not remove RSA keys from the router. To immediately remove RSA keys from the router, set the timeout value to zero.

**Examples**

The following example shows how to set the time that the router will wait before removing the RSA keys that are stored in the eToken after the eToken has been removed from the router:

```
crypto pki token usbtoken0 removal timeout 60
```

**Related Commands**

Command	Description
<b>crypto pki token logout</b>	Logs the router out of the USB token.
<b>crypto pki token max-retries</b>	Sets the maximum number of allowed failed login attempts.

# crypto pki token secondary config

To merge a specified file with the running configuration after the eToken is logged into the router, use the **crypto pki token secondary config** command in privileged EXEC mode.

**crypto pki token** *token-name* **secondary config** *file*

Syntax Description	<i>token-name</i>	Name of USB eToken that will have its running configuration merged with the secondary configuration file.
	<i>file</i>	Name of the file that will be merged with the running configuration.
	<b>Note</b>	The filename is relative to the eToken, so the name should not include a device name such as “usbtoken0:.”

**Defaults** A secondary configuration file does not exist.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use the **crypto pki token secondary config** command if you want to merge, not overwrite, a file with the running configuration on the router.

The secondary configuration is processed after the eToken is logged into the router.

**Examples** The following example shows how to merge the secondary configuration file “CONFIG1.CFG” with the current running configuration:

```
crypto pki token default secondary config CONFIG1.CFG
```

Related Commands	Command	Description
	<b>crypto pki token login</b>	Logs into the USB eToken.
	<b>crypto pki token user-pin</b>	Creates a PIN that automatically allows the router to log into the USB eToken at router startup.

# crypto pki token user-pin

To create a PIN that automatically allows the router to log into the USB eToken at router startup, use the **crypto pki token user-pin** command in global configuration mode. To remove the stored PIN from the configuration, use the **no** form of this command.

**crypto pki token** *token-name* **user-pin** [*pin*]

**no crypto pki token** *token-name* **user-pin** [*pin*]

Syntax Description	<i>token-name</i>	Name of USB eToken that the router will log into.
	<i>pin</i>	(Optional) User PIN required to log into the eToken. The PINs are stored in private NVRAM. If a user PIN is not specified, the default PIN, 1234567890, will be used.

**Defaults** If this command is not issued, the router cannot access the eToken.

**Command Modes** Global configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** After the eToken is plugged into the router, the router will use the specified PIN (or the default PIN if no PIN is specified) to automatically log in as the user.

**Examples** The following example shows how to access the eToken via the user pin “12345”:

```
crypto pki token usbtokens0 user-pin 12345
```

Related Commands	Command	Description
	<b>crypto pki login</b>	Logs into the USB eToken.
	<b>crypto pki token logout</b>	Logs the router out of the USB eToken.

# debug usb driver

To display debug messages about USB transfers, use the **debug usb driver** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug usb driver** [*transfer transfer-method*]

**no debug usb driver** [*transfer transfer-method*]

<b>Syntax Description</b>	<b>transfer</b>	(Optional) Specifies the type of transfer method for which messages are to be displayed on the console.
	<i>transfer-method</i>	One of the following options: <b>interrupt</b> , <b>bulk</b> , or <b>control</b> .

**Command Default** None

**Command Modes** Privileged EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
		12.3(14)T

**Usage Guidelines** The **debug usb driver** command produces a large amount of data that might slow down your system, so use this command with caution.

**Examples** The following sample debug output is produced when the the **debug usb driver** command with the **transfer** and **control** keywords is issued and when an eToken is unplugged and plugged back in:

```
Router# debug usb driver transfer bulk

USB Driver Bulk Transfer debugging is on
Router# debug usb driver transfer control

USB Driver Control Transfer debugging is on

Router# debug usb stack

Stack debugging is on
Router#
Router#
*Dec 22 06:18:29.399:%USB_HOST_STACK-6-USB_DEVICE_DISCONNECTED:A USB device has been
removed from port 1.
*Dec 22 06:18:29.499:Detached:
*Dec 22 06:18:29.499:Host:          1
*Dec 22 06:18:29.499:Address:      18
*Dec 22 06:18:29.499:Manufacturer: AKS
*Dec 22 06:18:29.499:Product:      eToken Pro 4254
*Dec 22 06:18:29.499:Serial Number:
Router#
```

```
*Dec 22 06:18:29.499:%USB_TOKEN_FILESYS-6-USB_TOKEN_REMOVED:USB Token device
removed:usbtoken1.
*Dec 22 06:18:29.499:%CRYPTO-6-TOKENREMOVED:Cryptographic token eToken removed from
usbtoken1
Router#
Router#
Router#
Router#
Router#
*Dec 22 06:18:38.063:%USB_HOST_STACK-6-USB_DEVICE_CONNECTED:A Low speed USB device has
been inserted in port 1.
*Dec 22 06:18:38.683:ATTACHED====>Class-driver activated
*Dec 22 06:18:38.683:Host:          1
*Dec 22 06:18:38.683:Address:      19
*Dec 22 06:18:38.683:Manufacturer: AKS
*Dec 22 06:18:38.683:Product:      eToken Pro 4254
*Dec 22 06:18:38.683:Serial Number:
*Dec 22 06:18:39.383:Control Transfer
Device Handle:0x3010000
Direction:0x0
Request:0x1
Type:0x40
Recipient:0x0
ValueDesc:0x0
ValueIndex:0x0
Index:0x0

*Dec 22 06:18:39.383:Control Transfer
Device Handle:0x3010000
Direction:0x80
Request:0x81
Type:0x40
Recipient:0x0
ValueDesc:0x0
ValueIndex:0x0
Index:0x0

*Dec 22 06:18:39.407:Control Transfer
Device Handle:0x3010000
Direction:0x0
Request:0x3
Type:0x40
Recipient:0x0
ValueDesc:0x0
ValueIndex:0x0
Index:0x0

*Dec 22 06:18:39.407:Control Transfer
Device Handle:0x3010000
Direction:0x80
Request:0
my3825#x83
Type:0x40
Recipient:0x0
ValueDesc:0x0
ValueIndex:0x0
Index:0x0

*Dec 22 06:18:39.503:Control Transfer
Device Handle:0x3010000
Direction:0x0
Request:0x2
Type:0x40
Recipient:0x0
```

ValueDesc:0x0  
ValueIndex:0x0  
Index:0x0

\*Dec 22 06:18:39.507:Control Transfer  
Device Handle:0x3010000  
Direction:0x80  
Request:0x82  
Type:0x40  
Recipient:0x0  
ValueDesc:0x0  
ValueIndex:0x0  
Index:0x0

\*Dec 22 06:18:39.507:%USB\_TOKEN\_FILESYS-6-USB\_TOKEN\_INSERTED:USB Token device  
inserted:usbtoken1.

\*Dec 22 06:18:39.515:Control Transfer  
Device Handle:0x3010000  
Direction:0x0  
Request:0x6  
Type:0x40  
Recipient:0x0  
ValueDesc:0x0  
ValueIndex:0x0  
Index:0x0

\*Dec 22 06:18:39.515:%USB\_TOKEN\_FILESYS-6-REGISTERING\_WITH\_IFS:Registering USB Token File  
System usbtoken1:might take a while...

\*Dec 22 06:18:39.515:Control Transfer  
Device Handle:0x3010000  
Direction:0x80  
Request:0x86  
Type:0x40  
Recipient:0x0  
ValueDesc:0x0  
ValueIndex:0x0  
Index:0x0

\*Dec 22 06:18:39.543:Control Transfer  
Device Handle:0x3010000  
Direction:0x0  
Request:0x6  
Type:0x40  
Recipient:0x0  
ValueDesc:0x0  
ValueIndex:0x0  
Index:0x0

.  
.  
.

# delete

To delete a file on a Flash memory device or NVRAM, use the **delete** command in EXEC mode.

**delete** *url* [/force | /recursive]

Syntax Description		
<i>url</i>		Cisco IOS File System URL of the file to be deleted. Include the file system prefix, followed by a colon, and, optionally, the name of a file or directory. See <a href="#">Table 7</a> for list of supported URLs.
<b>/force</b>		(Optional) Deletes the specified file or directory without prompting you for verification.  <b>Note</b> Use this keyword with caution: the system will not ask you to confirm the file deletion.
<b>/recursive</b>		(Optional) Deletes all files in the specified directory, as well as the directory itself.

**Command Modes** EXEC

Command History	Release	Modification
	11.0	This command was introduced.
	12.3(14)T	The <b>usbflash[0-9]:</b> and <b>usbtoken[0-9]:</b> options were added to the list of Cisco IOS File System URLs.

**Usage Guidelines** If you attempt to delete the configuration file or image specified by the CONFIG\_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

When you delete a file in Flash memory, the software simply marks the file as deleted, but it does not erase the file. To later recover a “deleted” file in Flash memory, use the **undelete** EXEC command. You can delete and undelete a file up to 15 times.

To permanently delete all files marked “deleted” on a linear Flash memory device, use the **squeeze** EXEC command.

[Table 7](#) contains a list of Cisco IOS File System URLs.

**Table 7** URL File System Prefix Keywords

Prefix	Filesystem
<b>bootflash:</b>	Delete the file from boot Flash memory.
<b>flash:</b>	Delete the file from Flash memory.
<b>nvrाम:</b>	Delete the from the router NVRAM.
<b>slot0:</b>	Delete the file from the first PCMCIA Flash memory card.

**Table 7** *URL File System Prefix Keywords (Continued)*

Prefix	Filesystem
<b>usbflash[0-9]:</b>	Delete the file from the USB Flash drive.
<b>usbtoken[0-9]:</b>	Delete the file from the USB eToken.

### Examples

The following example deletes the file named test from the Flash card inserted in slot 0:

```
Router# delete slot0:test
Delete slot0:test? [confirm]
```

### Related Commands

Command	Description
<b>cd</b>	Changes the default directory or file system.
<b>dir</b>	Displays a list of files on a file system.
<b>show bootvar</b>	Displays the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, the contents of the BOOTLDR environment variable, and the configuration register setting.
<b>squeeze</b>	Permanently deletes Flash files by squeezing a Class A Flash file system.
<b>undelete</b>	Recovers a file marked “deleted” on a Class A or Class B Flash file system.

# dir

To display a list of files on a file system, use the **dir** command in EXEC mode.

**dir** [/all] [filesystem: ][file-url]

Syntax Description		
/all	(Optional)	Lists deleted files, undeleted files, and files with errors.
filesystem:	(Optional)	File system or directory containing the files to list, followed by a colon.
file-url	(Optional)	The name of the files to display on a specified device. The files can be of any type. You can use wildcards in the filename. A wildcard character (*) matches all patterns. Strings after a wildcard are ignored.

**Defaults** The default file system is specified by the **cd** command. When you omit the /all keyword, the Cisco IOS software displays only undeleted files.

**Command Modes** EXEC

Command History	Release	Modification
	11.0	This command was introduced.
	12.3	A timestamp that shows the offset from Coordinated Universal Time (UTC) was added to the <b>dir</b> command display.
	12.3(14)T	The <b>usbflash[0-9]:</b> and <b>usbtoken[0-9]:</b> options were added as available file systems.

**Usage Guidelines** Use the **show** (Flash file system) command to display more detail about the files in a particular file system.

**Examples** The following is sample output from the **dir** command:

```
Router# dir slot0:

Directory of slot0:/

 1  -rw-      4720148  Dec 29 2003 17:49:36 -08:00 hampton/nitro/c7200-j-mz
 2  -rw-      4767328   Jan 02 2004 18:42:53 -08:00 c7200-js-mz
 5  -rw-         639   Jan 03 2004 12:09:32 -08:00 rally
 7  -rw-         639   Jan 03 2004 12:37:13 -08:00 the_time

20578304 bytes total (3104544 bytes free)

Router# dir /all slot0:
```

Directory of slot0:/

```

1  -rw-      4720148  Dec 15 2003 17:49:36 -08:00 hampton/nitro/c7200-j-mz
2  -rw-      4767328   Jan 02 2004 18:42:53 -08:00 c7200-js-mz
3  -rw-      7982828   Jan 02 2004 18:48:14 -08:00 [rsp-jsv-mz]
4  -rw-         639   Jan 03 2004 12:09:17 -08:00 the_time]
5  -rw-         639   Jan 03 1994 12:09:32 -08:00 rally
6  -rw-         639   Jan 03 1994 12:37:01 -08:00 [the_time]
7  -rw-         639   Jan 03 1994 12:37:13 -08:00

```

Table 8 describes the significant fields shown in the output.

**Table 8** *dir Field Descriptions*

Field	Description
1	Index number of the file.
-rw-	Permissions. The file can be any or all of the following: <ul style="list-style-type: none"> <li>• d—directory</li> <li>• r—readable</li> <li>• w—writable</li> <li>• x—executable</li> </ul>
4720148	Size of the file.
Dec 15 2003 17:49:36	Last modification date.
-08:00	Conversion to local time in hours from Coordinated Universal Time (UTC). In the example, -08:00 indicates that the given time is 8 hours behind UTC or Pacific Standard Time (PST).
hampton/nitro/c7200-j-mz	Filename. Deleted files are indicated by square brackets around the filename.

#### Related Commands

Command	Description
<b>cd</b>	Changes the default directory or file system.
<b>delete</b>	Deletes a file on a Flash memory device.
<b>undelete</b>	Recovers a file marked “deleted” on a Class A or Class B Flash file system.

# format

To format a Class A, Class B, or Class C Flash file system, use the **format** command in EXEC mode.

## Class B and Class C Flash File Systems

**format** *filesystem1*:

## Class A Flash File System

**format** [**spare** *spare-number*] *filesystem1*: [[*filesystem2*:][*monlib-filename*]]



### Caution

Reserve a certain number of memory sectors as spares, so that if some sectors fail, most of the Flash memory card can still be used. Otherwise, you must reformat the Flash card when some of the sectors fail.

### Syntax Description

<b>spare</b>	(Optional) Reserves spare sectors as specified by the <i>spare-number</i> argument when formatting Flash memory.
<i>spare-number</i>	(Optional) Number of the spare sectors to reserve on formatted Flash memory. Valid values are from 0 to 16. The default value is zero.
<i>filesystem1</i> :	Flash memory to format, followed by a colon.
<i>filesystem2</i> :	(Optional) File system containing the monlib file to use for formatting <i>filesystem1</i> followed by a colon.
<i>monlib-filename</i>	(Optional) Name of the ROM monitor library file (monlib file) to use for formatting the <i>filesystem1</i> argument. The default monlib file is the one bundled with the system software.  When used with HSA and you do not specify the <i>monlib-filename</i> argument, the system takes ROM monitor library file from the slave image bundle. If you specify the <i>monlib-filename</i> argument, the system assumes that the files reside on the slave devices.

### Command Default

None

### Command Modes

EXEC

### Command History

Release	Modification
11.0	This command was introduced.
12.3(14)T	Support for Class B Flash (USB Flash and USB eToken) File Systems was added.

**Usage Guidelines**

Use this command to format Class A, B, or C Flash memory file systems.

In some cases, you might need to insert a new PCMCIA Flash memory card and load images or backup configuration files onto it. Before you can use a new Flash memory card, you must format it.

Sectors in Flash memory cards can fail. Reserve certain Flash memory sectors as “spares” by using the optional *spare* argument on the **format** command to specify 0 to 16 sectors as spares. If you reserve a small number of spare sectors for emergencies, you can still use most of the Flash memory card. If you specify 0 spare sectors and some sectors fail, you must reformat the Flash memory card, thereby erasing all existing data.

The monlib file is the ROM monitor library. The ROM monitor uses this file to access files in the Flash file system. The Cisco IOS system software contains a monlib file.

In the command syntax, *filesystem1*: specifies the device to format and *filesystem2*: specifies the optional device containing the monlib file used to format *filesystem1*:. If you omit the optional *filesystem2*: and *monlib-filename* arguments, the system formats *filesystem1*: using the monlib file already bundled with the system software. If you omit only the optional *filesystem2*: argument, the system formats *filesystem1*: using the monlib file from the device you specified with the **cd** command. If you omit only the optional *monlib-filename* argument, the system formats *filesystem1*: using the *filesystem2*: monlib file. When you specify both arguments—*filesystem2*: and *monlib-filename*—the system formats *filesystem1*: using the monlib file from the specified device. You can specify *filesystem1*:’s own monlib file in this argument. If the system cannot find a monlib file, it terminates its formatting.

**Note**

You can read from or write to Flash memory cards formatted for Cisco 7000 series Route Processor (RP) cards in your Cisco 7200 and 7500 series routers, but you cannot boot the Cisco 7200 and 7500 series routers from a Flash memory card formatted for the Cisco 7000 series routers. Similarly, you can read from or write to Flash memory cards formatted for the Cisco 7200 and 7500 series routers in your Cisco 7000 series routers, but you cannot boot the Cisco 7000 series routers from a Flash memory card formatted for the Cisco 7200 and 7500 series routers.

**Examples**

The following example formats a Flash memory card inserted in slot 0:

```
Router# format slot0:

Running config file on this device, proceed? [confirm]y
All sectors will be erased, proceed? [confirm]y
Enter volume id (up to 31 characters): <Return>
Formatting sector 1 (erasing)
Format device slot0 completed
```

When the console returns to the EXEC prompt, the new Flash memory card is formatted and ready for use.

**Related Commands**

Command	Description
<b>cd</b>	Changes the default directory or file system.
<b>copy</b>	Copies any file from a source to a destination.
<b>delete</b>	Deletes a file on a Flash memory device.
<b>show file systems</b>	Lists available file systems.
<b>squeeze</b>	Permanently deletes Flash files by squeezing a Class A Flash file system.
<b>undelete</b>	Recovers a file marked “deleted” on a Class A or Class B Flash file system.

# show usb controllers

To display USB host controller information, use the **show usb controllers** command in Privileged EXEC mode.

**show usb controllers** [*controller-number*]

<b>Syntax Description</b>	<i>controller-number</i> (Optional) Displays information only for the specified controller.
---------------------------	---

<b>Defaults</b>	Information about all controllers on the system are displayed.
-----------------	--

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

Command History	Release	Modification
	12.3(14)T	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>show usb controllers</b> command to display content such as controller register specific information, current asynchronous buffer addresses, and period scheduling information. You can also use this command to verify that copy operations are occurring successfully onto a USB flash module.
-------------------------	---

<b>Examples</b>	The following example is sample output from the <b>show usb controller</b> command:
-----------------	---

```
Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
  Revision:0x11
  Control:0x80
  Command Status:0x0
  Hardware Interrupt Status:0x24
  Hardware Interrupt Enable:0x80000040
  Hardware Interrupt Disable:0x80000040
  Frame Interval:0x27782EDF
  Frame Remaining:0x13C1
  Frame Number:0xDA4C
  LSThreshold:0x628
  RhDescriptorA:0x19000202
  RhDescriptorB:0x0
  RhStatus:0x0
  RhPort1Status:0x100103
  RhPort2Status:0x100303
  Hardware Configuration:0x3029
  DMA Configuration:0x0
  Transfer Counter:0x1
  Interrupt:0x9
  Interrupt Enable:0x196
  Chip ID:0x3630
```

Buffer Status:0x0  
 Direct Address Length:0x80A00  
 ATL Buffer Size:0x600  
 ATL Buffer Port:0x0  
 ATL Block Size:0x100  
 ATL PTD Skip Map:0xFFFFFFFF  
 ATL PTD Last:0x20  
 ATL Current Active PTD:0x0  
 ATL Threshold Count:0x1  
 ATL Threshold Timeout:0xFF

Int Level:1

Transfer Completion Codes:

Success	:920	CRC	:0
Bit Stuff	:0	Stall	:0
No Response	:0	Overrun	:0
Underrun	:0	Other	:0
Buffer Overrun	:0	Buffer Underrun	:0

Transfer Errors:

Canceled Transfers	:2	Control Timeout	:0
--------------------	----	-----------------	----

Transfer Failures:

Interrupt Transfer	:0	Bulk Transfer	:0
Isochronous Transfer	:0	Control Transfer	:0

Transfer Successes:

Interrupt Transfer	:0	Bulk Transfer	:26
Isochronous Transfer	:0	Control Transfer	:894

USB D Failures:

Enumeration Failures	:0	No Class Driver Found	:0
Power Budget Exceeded	:0		

USB MSCD SCSI Class Driver Counters:

Good Status Failures	:3	Command Fail	:0
Good Status Timed out	:0	Device not Found	:0
Device Never Opened	:0	Drive Init Fail	:0
Illegal App Handle	:0	Bad API Command	:0
Invalid Unit Number	:0	Invalid Argument	:0
Application Overflow	:0	Device in use	:0
Control Pipe Stall	:0	Malloc Error	:0
Device Stalled	:0	Bad Command Code	:0
Device Detached	:0	Unknown Error	:0
Invalid Logic Unit Num	:0		

USB Aladdin Token Driver Counters:

Token Inserted	:1	Token Removed	:0
Send Insert Msg Fail	:0	Response Txns	:434
Dev Entry Add Fail	:0	Request Txns	:434
Dev Entry Remove Fail	:0	Request Txn Fail	:0
Response Txn Fail	:0	Command Txn Fail	:0
Txn Invalid Dev Handle	:0		

USB Flash File System Counters:

Flash Disconnected	:0	Flash Connected	:1
Flash Device Fail	:0	Flash Ok	:1
Flash startstop Fail	:0	Flash FS Fail	:0

USB Secure Token File System Counters:

Token Inserted	:1	Token Detached	:0
Token FS success	:1	Token FS Fail	:0
Token Max Inserted	:0	Create Talker Failures	:0
Token Event	:0	Destroy Talker Failures	:0
Watched Boolean Create Failures	:0		

## show usb device

To display USB device information, use the **show usb device** command in privileged EXEC mode.

**show usb device** [*controller-ID* [*device-address*]]

Syntax Description		
<i>controller-ID</i>	(Optional)	Displays information only for the devices under the specified controller.
<i>device-address</i>	(Optional)	Displays information only for the device with the specified address.

**Defaults** Information for all devices attached to the system are displayed.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use the **show usb device** command to display information for either a USB flash drive or a USB eToken, as appropriate.

**Examples** The following example is sample output from the **show usb device** command:

```
Router# show usb device

Host Controller:1
Address:0x1
Device Configured:YES
Device Supported:YES
Description:DiskOnKey
Manufacturer:M-Sys
Version:2.0
Serial Number:0750D84030316868
Device Handle:0x1000000
USB Version Compliance:2.0
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x8EC
Product ID:0x15
Max. Packet Size of Endpoint Zero:64
Number of Configurations:1
Speed:Full
Selected Configuration:1
Selected Interface:0
```

```
Configuration:
  Number:1
  Number of Interfaces:1
  Description:
  Attributes:None
  Max Power:140 mA

  Interface:
    Number:0
    Description:
    Class Code:8
    Subclass:6
    Protocol:80
    Number of Endpoints:2

    Endpoint:
      Number:1
      Transfer Type:BULK
      Transfer Direction:Device to Host
      Max Packet:64
      Interval:0

    Endpoint:
      Number:2
      Transfer Type:BULK
      Transfer Direction:Host to Device
      Max Packet:64
      Interval:0

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
  Number:1
  Number of Interfaces:1
  Description:
  Attributes:None
  Max Power:60 mA

  Interface:
    Number:0
    Description:
    Class Code:255
    Subclass:0
    Protocol:0
    Number of Endpoints:0
```

Table 9 describes the significant fields shown in the display.

**Table 9** *show usb device Field Descriptions*

Field	Description
Device handle	Internal memory handle allocated to the device.
Device Class code	The class code supported by the device. This number is allocated by the USB-IF. If this field is reset to 0, each interface within a configuration specifies its own class information, and the various interfaces operate independently. If this field is set to a value between 1 and FEH, the device supports different class specifications on different interfaces, and the interfaces may not operate independently. This value identifies the class definition used for the aggregate interfaces. If this field is set to FFH, the device class is vendor-specific.
Device Subclass code	The subclass code supported by the device. This number is allocated by the USB-IF.
Device Protocol	The protocol supported by the device. If this field is set to 0, the device does not use class-specific protocols on a device basis. If this field is set to 0xFF, the device uses a vendor-specific protocol on a device basis.
Interface Class code	The class code supported by the interface. If the value is set to 0xFF, the interface class is vendor specific. All other values are allocated by the USB-IF.
Interface Subclass code	The subclass code supported by the interface. All values are allocated by the USB-IF.
Interface Protocol	The protocol code supported by the interface. If this field is set to 0, the device does not use a class-specific protocol on this interface. If this field is set to 0xFF, the device uses a vendor-specific protocol for this interface.
Max Packet	Maximum data packet size, in bytes.

# show usb driver

To display information about registered USB class drivers and vendor-specific drivers, use the **show usb driver** command in privileged EXEC mode.

**show usb driver** [*index*]

<b>Syntax Description</b>	<i>index</i> (Optional) Displays information only for drivers on the specified index.				
<b>Defaults</b>	Information about all drivers is displayed.				
<b>Command Modes</b>	Privileged EXEC				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>12.3(14)T</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	12.3(14)T	This command was introduced.
Release	Modification				
12.3(14)T	This command was introduced.				

## Examples

The following example is sample output for the **show usb driver** command:

```
Router# show usb driver

Index:0
Owner Mask:0x6
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x8
Interface Subclass Code:0x6
Interface Protocol Code:0x50
Product ID:0x655BD598
Vendor ID:0x64E90000
Attached Devices:
    Controller ID:1, Device Address:1

Index:1
Owner Mask:0x1
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x0
Interface Subclass Code:0x0
Interface Protocol Code:0x0
Product ID:0x514
Vendor ID:0x529
Attached Devices:
    Controller ID:1, Device Address:17

Index:2
Owner Mask:0x5
Class Code:0x9
Subclass Code:0x6249BD58
```

```

Protocol:0x2
Interface Class Code:0x5DC0
Interface Subclass Code:0x5
Interface Protocol Code:0xFFFFFFFF
Product ID:0x2
Vendor ID:0x1
Attached Devices:
    None

Index:3
Owner Mask:0x10
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x0
Interface Subclass Code:0x0
Interface Protocol Code:0x0
Product ID:0x0
Vendor ID:0x0
Attached Devices:
    None
    
```

Table 10 describes the significant field shown in the display.

**Table 10** *show usb driver Field Descriptions*

Field	Description
Owner Mask	Indicates the fields that are used in enumeration comparison. The driver can own different devices on the basis of their product or vendor IDs and device or interface class, subclass, and protocol codes.

# show usb port

To display USB root hub port information, use the **show usb port** command in privileged EXEC mode.

**show usb port** [*port-number*]

<b>Syntax Description</b>	<i>port-number</i>	(Optional) Displays information only for a specified. If the <i>port-number</i> is not issued, information for all root ports will be displayed.
---------------------------	--------------------	--

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.3(14)T	This command was introduced.

**Examples** The following sample from the **show usb port** command shows the status of the port 1 on the router:

```
Router# show usb port

Port Number:0
Status:Enabled
Connection State:Connected
Speed:Full
Power State:ON

Port Number:1
Status:Enabled
Connection State:Connected
Speed:Low
Power State:ON
```

# show usbtoken

To display information about the USB eToken (such as the eToken ID), use the **show usbtoken** command in privileged EXEC mode.

**show usbtoken**[0-9]:[all | *filesystem*]

Syntax Description	0-9	(Optional) One of the ten available flash drives you can choose from; valid values: 0-9. If you do not specify a number, 0 is used by default
	<b>all</b>	(Optional) All configuration files stored on the eToken.
	<i>filesystem</i>	(Optional) Name of a configuration file.

**Command Default** None

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use the **show usbtoken** command to verify whether a USB eToken is inserted in the router.

**Examples** The following example is sample output from the **show usbtoken** command:

```
Router# show usbtoken0

Token ID           :43353334
Token device name  : token0
Vendor name        : Aladdin
Product Name       : Etoken Pro
Serial number      : 22273a334353
Firmware version   : 4.1.3.2
Total memory size  : 32 KB
Free memory size   : 16 KB
FIPS version       : Yes/No
Token state        : "Active" | "User locked" | "Admin locked" | "System Error" |
                    "Unknown"
ATR (Answer To Reset) : "3B F2 98 0 FF C1 10 31 FE 55 C8 3"
```

Table 11 describes the significant fields shown in the display.

**Table 11** *show usbtoken Field Descriptions*

Field	Description
Token ID	Token identifier.
Token device name	A unique name derived by the token driver.
ATR (Answer to Reset)	Information replied by Smart cards when a reset command is issued.

# show usb tree

To display information about the port state and all attached devices, use the **show usb tree** command in privileged EXEC mode.

## show usb tree

**Syntax Description** This command has no arguments or keywords.

**Command Default** None

**Command Modes** EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Examples** The following example is sample output from the **show usb tree** command. This output shows that both a USB flash module and a USB eToken are currently enabled.

```
Router# show usb tree

[Host Id:1, Host Type:1362HCD, Number of RH-Port:2]
<Root Port0:Power=ON      Current State=Enabled>
  Port0: (DiskOnKey) Addr:0x1 VID:0x08EC PID:0x0015 Configured (0x1000000)
<Root Port1:Power=ON      Current State=Enabled>
  Port1: (eToken Pro 4254) Addr:0x11 VID:0x0529 PID:0x0514 Configured (0x1010000)
```

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Copyright © 2005 Cisco Systems, Inc. All rights reserved.