



Storing Data In USB

The Universal Serial Bus (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.

- [Finding Feature Information, page 1](#)
- [Prerequisites for Storing Data In USB, page 1](#)
- [Restrictions for Storing Data In USB, page 2](#)
- [Information About Storing Data In USB, page 2](#)
- [How to Set Up and Use USB Modules on Cisco Routers, page 5](#)
- [Configuration Examples for Secure Token Support, page 15](#)
- [Additional References, page 16](#)
- [Feature Information for Storing Data In USB, page 18](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Storing Data In USB

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 Storing Data In USB

- 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 Storing Data In USB

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

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

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 **cryptokitokenremovaltimeout** command.)

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 **bootconfig** command to specify the new configuration located on the USB flash device; for example **bootconfigusbflash0: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, the table below 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"> • 64MB • 128MB • 256MB
File Types	<ul style="list-style-type: none"> • Typically used to store digital certificates, preshared keys, and router configurations for IPSec VPNs. • eTokens cannot store Cisco IOS images. 	Stores a file type that might be stored on a compact flash.
Security	<ul style="list-style-type: none"> • Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format. 	Files can be stored only in a nonsecure format.

Function	USB eToken	USB Flash
Boot Configurations	<ul style="list-style-type: none"> • The router can use the configuration stored in the eToken during boot time • The router can use the secondary configuration stored in the eToken during boot time. (A secondary configuration allows users to load their IPSec configuration.) 	<ul style="list-style-type: none"> • Configuration file can be automatically transferred from the USB Flash to the router if the bootconfig command is issued (for example, bootconfigusbflash0:new-config).

USB Storage Filesystem Support

Since USB storage device capacities are increasing, it is necessary that the DOSFS and the usbflash components are modified so that large capacity USB storage devices can be used. The USB Storage Filesystem Support feature extends DOSFS support for USB flash devices. With this feature you can use large capacity USB storage devices for data storage.

Benefits of Storing Data In USB

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.

Login Methods for the eToken

AutomaticLogin

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 the user to know the actual token PIN. The Aladdin's Windows-based utilities can be used to copy the RSA keys and secondary config files from the eToken if the user has physical access to the eToken.

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

To store the configuration file in the USB flash drive module or in an eToken, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot config** *file-system-prefix* : [*directory/*]*filename* [*nvbypass*]

DETAILED STEPS

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

Accessing and Setting Up the eToken

Logging Into the eToken

To log into an eToken manually or automatically, complete the tasks in this section.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
 - **crypto pki token token-name** [admin] login [pin]
3. **crypto pki token token-name user-pin** [pin]
4. **exit**
5. **show usbtokens 0-9** : *filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router> enable</pre>	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>Do one of the following:</p> <ul style="list-style-type: none"> crypto pki token token-name [admin] login [pin] <p>Example:</p> <pre>Router# crypto pki token usbtoken0 admin login 5678</pre> <p>Example:</p> <p style="text-align: center;">configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Manually logs into the eToken.</p> <p>You must specify the admin keyword if later you want to change the user PIN.</p> <p>or</p> <p>Puts the router in global configuration mode, which allows you to configure automatic eToken login.</p>
Step 3	<pre>crypto pki token token-name user-pin [pin]</pre> <p>Example:</p> <pre>Router(config)# crypto pki token usbtoken0 user-pin 1234</pre>	<p>(Optional) Creates a PIN that automatically allows the router to log into the USB eToken at router startup.</p> <p>Note Do not issue this command if you have already set up manual login.</p>
Step 4	<pre>exit</pre> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode.</p>
Step 5	<pre>show usbtoken 0-9 : filename</pre> <p>Example:</p> <pre>Router#</pre>	<p>(Optional) Verifies whether the USB eToken has been logged onto the router.</p>

What to Do Next

- 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.

Setting Administrative Functions on the eToken

To change default settings, such as the user PIN and the maximum number of failed on the eToken, complete the tasks in this section:

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** [seconds]
5. **crypto pki token** {*token-name* | default} **max-retries** [number]
6. **exit**
7. **copy usbflash** [**09**:*filename**destination-url*]
8. **show usbtokens** **0-9** : *filename*
9. **crypto pki token** *token-name* **logout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	crypto pki token <i>token-name</i> admin] change-pin [<i>pin</i>] Example: Router# crypto pki token usbtokens0 admin change-pin	(Optional) Changes the user PIN number on the USB eToken. <ul style="list-style-type: none"> • If the PIN is not changed, the default PIN--1234567890--will be used. Note After the PIN has been changed, you must reset the login failure count to zero (via the cryptopkitokenmax-retries command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 4	crypto pki token { <i>token-name</i> default} removal timeout [seconds] Example: Router(config)# crypto pki token usbtokens0 removal timeout 60	(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. Note 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.

	Command or Action	Purpose
Step 5	<pre>crypto pki token {token-name default} max-retries [number]</pre> <p>Example:</p> <pre>Router(config)# crypto pki token usbtoken0 max-retries 20</pre>	<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"> By default, the value is set at 15.
Step 6	<pre>exit</pre> <p>Example:</p> <pre>Router(config)# exit</pre>	Exits global configuration mode.
Step 7	<pre>copy usbflash [09;filename]destination-url</pre> <p>Example:</p> <pre>Router# copy usbflash0:</pre>	<p>Copies files from the router to the eToken.</p> <ul style="list-style-type: none"> <i>destination-url</i> --See the copy command page documentation for a list of supported options.
Step 8	<pre>show usbtokens 0-9 : filename</pre> <p>Example:</p> <pre>Router#</pre>	(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	<pre>crypto pki token token-name logout</pre> <p>Example:</p> <pre>Router# crypto pki token usbtoken0 logout</pre>	<p>Logs the router out of the USB eToken.</p> <p>Note If you want to save any data to the USB eToken, you must log back into the eToken.</p>

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 showfileystems Command

SUMMARY STEPS

1. Use the **showfileystems** 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:
2. Use the **showfileystems** 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 **showfileystems** command will display an error indicating an incompatible file system.

DETAILED STEPS

Step 1 Use the **showfileystems** 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 **showfileystems** 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 **showfileystems** command will display an error indicating an incompatible file system.

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

Example:

```
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

SUMMARY STEPS

1. Use the **showusbdevice** 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.

DETAILED STEPS

Use the **showusbdevice** 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:

Example:

```
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:

Example:

```
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

SUMMARY STEPS

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

DETAILED STEPS

Use the **showusbcontrollers** command to determine if there is a hardware problem with a USB Flash module. If the **showusbcontrollers** command displays an error, it indicates a hardware problem in the USB module.

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

Sample output for the **showusbcontrollers** command for a working USB Flash module appears below:

Example:

```
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
Device Stalled :0
Device Detached :0
Invalid Logic Unit Num: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
USB Flash File System Counters:
Flash Disconnected :0
Flash Device Fail :0
Flash startstop 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

Malloc Error :0
Bad Command Code:0
Unknown Error :0

Token Removed :0
Response Txns :434
Request Txns :434
Request Txn Fail:0
Command Txn Fail:0

Flash Connected :1
Flash Ok :1
Flash FS Fail :0

Token Detached :0
Token FS Fail :0
Create Talker Failures:0
Destroy Talker Failures:0

```

The dir Command

SUMMARY STEPS

1. Use the **dir** command with the **usbflash09:** or the **usbtoken09:** keyword to display all files, directories, and their permission strings on the USB Flash or USB eToken.

DETAILED STEPS

Use the **dir** command with the **usbflash09:** or the **usbtoken09:** 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:

Example:

```

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:

Example:

```

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:

Example:

```

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 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)

```

Configuration Examples for Secure Token Support

Example Logging Into and Saving RSA Keys to eToken

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
    0E30EFE6 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 **showcryptokeymypubkeyrsa** 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 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001

```

Additional References

The following sections provide references related to the Storing data using USB feature.

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	USB eToken and USB Flash Features Support
File management (loading, copying, and rebooting files)	The section “File Management” in the <i>Cisco IOS Configuration Fundamentals and Network Management Configuration Guide</i>
Configuring digital certificate encryption	The chapter “Configuring Certification Authority Interoperability” in the <i>Cisco IOS Security Configuration Guide</i>

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: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Storing Data In USB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Storing Data In USB

Feature Name	Releases	Feature Information
USB Storage	12.3(14)T	<p>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.</p> <p>The following commands were introduced or modified: 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 usbtokens, show usb tree, boot config, copy, delete, dir, format.</p>

Feature Name	Releases	Feature Information
USB Storage Filesystem Support	12.2(33)SRE	<p>The USB Storage Filesystem Support feature extends DOSFS support for USB flash devices. With this feature you can use large capacity USB storage devices for data storage.</p> <p>In 12.2(33)SRE, this feature was introduced on the Cisco 7200-NPE-G2.</p> <p>The following commands were introduced or modified: cd, verify, mkdir, fsck.</p>

