

Chapter 6

Managing the File System

Introduction	6-3
File Naming Conventions	6-3
Long Filenames	6-5
Working with Files	6-5
Built-In Editor	6-6
Using Wildcards	6-7
Sending Output to a Text File	6-8
Flash Memory	6-8
The Flash File System (FFS)	6-9
If You Clear Flash Memory Completely	6-10
Non-Volatile Storage (NVS)	6-11
Secure Digital (SD) Cards	6-12
CompactFlash (CFlash) Cards	6-12
The FAT File System	6-13
Command Reference	6-15
activate flash compaction	6-15
add cflash dir	6-16
add file	6-17
clear card totally	6-18
clear cflash	6-18
clear flash totally	6-19
clear nvs totally	6-19
copy	6-20
create file	6-22
delete cflash dir	6-23
delete file	6-23
delete nvs	6-25
disable cflash test	6-25
dump nvs	6-26
edit	6-27
enable cflash test	6-30
modify nvs	6-30
purge file translationtable	6-31
rename	6-32
reset card counters	6-33
reset file permanentredirect	6-33
set cflash dir	6-34
show card	6-35
show card counters	6-36
show cflash	6-39
show cflash counters	6-40

show cflash file	6-41
show cflash test	6-42
show ffile	6-44
show file	6-46
show file permanentredirect	6-49
show flash	6-50
show flash physical	6-52
show nvs	6-53
show nvs free	6-54

Introduction

The file system provides a consistent file-based interface to the physical memory devices that store data on the switch. Examples of memory devices are flash, non-volatile storage, secure digital cards, and CompactFlash. The file system allows data, such as product software, licence information, and configuration scripts, to be stored on the switch in files with descriptive names and manipulated with a consistent set of commands, regardless of where they are physically stored. The file system provides a single directory on each storage device on x900-24X switches. Directories for x900-48FE and AT-9900 switches are supported on CompactFlash but files cannot be run from CompactFlash and must first be loaded into NVS or flash.

File Naming Conventions

x900-24X Files are uniquely identified by names consisting of three parts. The format for these parts is:

`[device:] filename.ext`

where:

- *device* specifies the physical memory device where the file is stored.
- *filename* is the base name, and is a descriptive name for the file. Invalid characters are “ * / : < > ? \ | . Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ! # \$ % & ` () + , - . ; = @ [] ^ _ ‘ { } ~ and a space
- *filename* cannot be any of the following:
 - aux
 - con
 - com1 to com4
 - lpt1 to lpt4
 - nul
 - prn
- *ext* is a file name extension separated by a period. Valid characters are the same as for *filename* except that *ext* cannot contain a period.
- The combined length of *filename* and *ext* can be from 1 to 32 characters.
- If a file name contains spaces or an equals sign, it must be in double quotes.

For information about wildcards, see [“Using Wildcards” on page 6-7](#).

**x900-48FE and
AT9900**

Files are uniquely identified by names consisting of three parts. The format for these parts is:

```
[device:] filename.ext
```

where:

- *device* specifies the physical memory device where the file is stored.
- *filename* is the base name, and is a descriptive name for the file. It can be from 1 to 28 characters long. Invalid characters are * + = " \ [] ; : ? / , < > . Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ~ ' ! @ # \$ % ^ & () _ - { }
- *.ext* is a file name extension 1 to 3 characters long. When specified, it must be separated from *filename* by a period. Valid characters are:
 - uppercase and lowercase letters
 - digits
 - hyphen
- If a file name contains spaces or an equals sign, it must be in double quotes.
- If a colon is in the file name, the device parameter is ignored and the switch assumes that the file name includes the device name.

File extensions

Switches determine file types based on extensions. The following table explains some of the ones possible.

Extension	File Type/Function
acc	Accounting information
bin	Bootloader software image file
cfg	Configuration or boot script
core	Core memory dump
hlp	Help file
htm	HTML file used by the HTTP server
lic	Licence information
log	Log file
mds	Modem script
paz	Compressed patch
pkg	Product software package
rez	Compressed product release file
rnd	Random number data
rsc	GUI resource file
scp	Script
sec	SNMP engine information
txt	Generic text file
lfn	In long filename translation table. Valid for x900-48FE and AT-9900 switches only.

Wildcards For information about wildcards, see [“Using Wildcards” on page 6-7](#).

Examples The following are examples of valid file names:

```
config.scp
card:setup.scp
"home office.log"
"test1=2.txt"
```

The following is an example of an illegal file name because the forward slash is not a valid delimiter for flash, and directories are not supported:

```
flash:/sys/head_o.cfg
```

Long Filenames

Software on x900-48FE and AT-9900 switches support long filenames in either DOS 16.3 or DOS 28.3 format. The following table summarises which software versions support different DOS filename formats.

Software release	Dos 8.3 format	DOS 16.3 format	DOS 28.3 format
2.4.x and earlier	Yes	No	No
2.5.1 and later	Yes	Yes	No
2.6.4 and later	Yes	Yes	Yes

Upgrading to new versions

When upgrading to this software version from earlier software releases, filenames retain their DOS naming format. DOS 8.3 format filenames remain in DOS 8.3 format and DOS 16.3 format filenames remain in DOS 16.3 format.

Back to previous versions

If you install a software version that supports DOS 28.3 format, and then you install a previous software version that supports **only** DOS 8.3 format, long filenames that were in DOS 28.3 format are truncated to DOS 8.3 format. When you reinstall a software version that supports DOS 28.3 format, these truncated filenames are restored to their DOS 28.3 format and no information is lost.

If you install a software version that supports DOS 28.3 format, and then you install a previous software version that supports DOS 16.3 format, long filenames in DOS 28.3 format are permanently truncated to DOS 8.3 format. For example, the file AB12345678.SCP is permanently renamed AB123~01.SCP. Any long filenames that were in DOS 28.3 format remain truncated in DOS 8.3 format when you reinstall a software version that supports DOS 28.3 format.

Working with Files

The x900-48FE and AT-9900 switches support file names that are up to 28 characters long and with 3-character extensions (DOS 28.3 format). However, the switches **store** files in the 8-character DOS 8.3 format with a 3-character extension. For example, the file named extralongfilenam.cfg can be saved as extral~1.cfg in the flash file system. Therefore, files can be accessed with two file names, either of which can be used for file management.

A translation table named longname.lfn converts file names between DOS 28.3 format and DOS 8.3 format. To reconcile file names, the switch checks this translation table, which is synchronised with file contents in memory. If the translation table is corrupted, it can be rebuilt from valid files in memory. To resynchronise the translation table to the file contents in memory, use the **purge file translationtable** command on page 6-31.

The **update** option restores valid long file names to the appropriate table entries after the table has been rebuilt. Long file names are deleted that are not reconciled to the new table. All table entries are deleted that are not confirmed to be in memory. This leaves a translation table that has maintained all of its previously valid data, and disposed of the rest. The table continues to support subsequent long file name creation and management.

The **all** option completely rebuilds the translation table. All long file names are lost. The table continues to support all subsequent long file name creation and management.

To display the contents of the translation table, which converts file names between DOS 28.3 format and DOS 8.3 format, use the command:

```
show file=longfile.lfn
```

To display files in the current directory on a CompactFlash card, use the command:

```
show cflash file
```

To display a directory of the files stored on the switch in both flash and NVS, use the command:

```
show file
```

To limit the display to certain files, use the command:

```
show file=filename
```

filename can contain wildcard characters * and |.

To permanently delete a file, use the command:

```
delete file=filename
```

filename can contain wildcard characters * and |. Note that you cannot use this command to delete the preferred software release or the current boot configuration file.

To create a text file, use the switch's built-in editor by using the command:

```
edit [filename]
```

To load a file onto a switch with HTTP, TFTP, or ZMODEM, use the command:

```
load file=filename
```

To change the name of a file, use the command:

```
rename src_filename dest_filename
```

To make a copy of a file with a new name or on a different storage device, use the command:

```
copy src_filename dest_filename
```

Built-In Editor

The switch has a built-in full-screen text editor for editing ASCII text files stored on the switch. You can use the editor to edit your current configuration file, or to create a script file that you can run manually or automatically.

The editor uses VT100 command sequences and should be used only with a VT100-compatible terminal, terminal emulation program, or Telnet client.

To start the editor with a new file or an existing file, enter the **edit** command on [page 6-27](#).

Using Wildcards

x900-24X Some file commands can process groups of files as well as single files. The asterisk (*) and vertical line (|) can be used as wildcard characters in these commands to identify a group of files or single files. The following table explains how wildcards are interpreted in each part of the file name, [device:]filename.ext.

For this part...	The asterisk can...	The vertical line can...
device	be used instead of a storage device name so that the command processes files on all storage devices	N/A
filename	match all files whose names start with the characters before the asterisk	match any single character
ext	match all files whose extensions start with the characters before the asterisk	match any single character

The following table has examples of valid wildcard expressions.

Example	Matches these files...
.	All files on the default storage device, flash.
:.txt	TXT files on all storage devices.
flash:sys*.core	CORE files on flash whose <i>filenames</i> start with "sys".
sys*abc.core	CORE files whose <i>filenames</i> start with "sys". All characters in the <i>filename</i> that follow the asterisk are ignored.
nvs: yz.s*	Files on NVS with 4-character filenames ending with "yz" and extensions starting with "s".

x900-48FE and AT-9900

Some file commands can process groups of files as well as single files. The asterisk (*) can be used as a wildcard character in these commands to identify a group of files.

The > character specifies a range of characters. For example a>z matches any letter in the alphabet.

The + character specifies a list of options. For example x*.scp+y*.scp would specify files that match x*.scp or y*.scp.

Square brackets specify a group of operations. For example, ppp*.[scp+cfg] matches scripts and configuration files whose names start with "PPP".

A vertical bar matches any single character. For example, | | |.scp matches script files with names three characters long (excluding extension and device name).

The following are examples of valid wildcard expressions:

```
flash:*. *
*:*.rez
```

Sending Output to a Text File

Instead of displaying command output on a screen, you may want to capture it in a text file in flash memory.

One time only

To send output from one or more individual commands to a text file, use the command:

```
create file=filename [command=commandstring]
other-parameters
```

To add output from other commands to the end of the file you created, use the command:

```
add file=filename command=commandstring
other-parameters
```

For example, to capture the output of the **show ip route** command in a text file called ip-route.txt, use the command:

```
create file=ip-route.txt command="show ip route"
```

To add output from the **show ip route count** command to the end of the ip-route.txt file, use the command:

```
add file=ip-route.txt command="show ip route count"
```

Ongoing

To capture ongoing output, use command:

```
create file=filename [command=commandstring]
[permanentredirect] other-parameters
```

For example, to enable IP route debugging and save the debug output in a file, first create the file using the **permanentredirect** parameter:

```
create file=ip-route.txt command="enable ip route debug" perm
```

When you have captured enough debug, or if you want to view the contents or upload the file, stop the capture by using the command:

```
reset file=ip-route.txt perm
```

Note that resetting the file does not disable the debugging, although the debug output is no longer displayed or saved.

To restart capturing output to the same file, use the command:

```
add file=ip-route.txt command="enable ip route debug" perm
```

Flash Memory

Flash memory allows the switch to store large volumes of data (up to 32 MBytes). It can store any type of file; product software, patches, and configuration files are stored in flash by default. Product software can be loaded into flash memory from a remote server over a switch port using the Loader functionality. Multiple files can be loaded and then individually selected at runtime by the Install functionality. Comprehensive management features are provided on x900-48FE and AT-9900 switches to examine the state of flash memory and to view or modify the contents.

To enable flash memory to support applications other than software releases on x900-48FE and AT-9900 switches, it is structured like a disk subsystem with files that can be created, deleted, read, and written by any switch module. Files can also be manipulated directly using the command line interface. This allows flash to be used to store any type of data, including releases, configurations, and logs.

Flash memory is non-volatile memory that can be erased and reprogrammed many times in situ. Flash memory has advantages over other types of non-volatile memory in that it has a very large storage capacity and does not require power from a battery to retain stored data.

A limitation of flash is that it has a fixed erase block size so that individual bytes cannot be changed without first clearing a whole block of data. Additionally, there is a limit on the number of erase cycles that can be done although the limit is quite high—typically at least 100,000 cycles. This would allow three erases per day for 100 years before the limit would be exceeded.

To display the amount of flash memory installed, use the [show system command on page 4-54 of Chapter 4, Configuring and Monitoring the System](#).

To display detailed information about flash memory on x900-48FE and AT-9900 switches, use the [show flash physical command on page 6-52](#).

See the Hardware Reference for the switch for more information about memory specifications.

The Flash File System (FFS)

The Flash File System (FFS) on x900-48FE and AT-9900 switches provides additional functionality to what the file system provides in order to manage the peculiarities of flash technologies. The additional functionality of the FFS includes:

- header and data integrity is ensured with a checksum mechanism.
- all flash processes can recover from a power cycle without data loss.
- automatic recovery of deleted file space by the compaction process.

Information about the state of the FFS can be displayed by using the [show flash command on page 6-50](#).

Working with FFS files

You can manage FFS files like other files on the switch by using the standard file system commands:

```
edit [filename]
delete file=filename
load=filename
show file [=filename]
```

In addition, you can use the following commands to manage files stored in flash memory. To display a directory of the files stored in flash memory, use the command:

```
show ffile [check]
```

If **check** is specified, the file data checksum is also verified. This is an option because it takes longer to complete a check on large files. A file data check is also carried out each time the system reads a file.

Compaction Flash memory has a granular erase structure that requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure, the FFS keeps track of the status of each file—whether it is being written, is complete, or is deleted.

The switch automatically compacts flash memory when a maximum threshold of deleted files is reached. Compaction searches through flash memory, copying good files to a new location. After the switch has copied the good files in an erase block, it clears the block. This creates space for new files by freeing up the space that was occupied by deleted files.

When a large amount of flash memory is in use, compaction may take several minutes. However, the switch continues to operate during the compaction process. A message appears when flash compaction begins; another one appears when it finishes.



Caution While flash is compacting, do not restart the switch or use commands that affect the flash file subsystem such as **create**, **edit**, **load**, **rename**, or **delete**. Wait until you get a message that file compaction is complete. Interrupting flash compaction may damages files.

Compaction can be manually initiated with the command:

```
activate flash compaction
```

FFS messages Some FFS processes generate messages in the system log (displayed with the [show log command on page 45-31 of Chapter 45, Logging Facility](#)) which include FFS message codes. See “Flash File System Message Codes” on [page B-7 of Appendix B, Reference Tables](#) for a list of codes and their meanings.

If You Clear Flash Memory Completely



Caution Do not completely clear flash memory. Files with product software, licence information, and install information are stored in flash and clearing it destroys them.

To recover from accidentally clearing flash

- 1. Boot up with default configuration.**

Reboot the switch from a terminal connected to the asynchronous terminal port (not Telnet). The switch uses fallback software to start on x900-24X switches. On x900-48FE and AT-9900 switches, use the install override to run the default configuration (see the Hardware Reference for more information).

- 2. Log in.**

Log into the switch by using the default password *friend* for the *manager* account.

- 3. Put current software version release on server.**

Make sure you have the current product software and patch files on a server connected to the switch. Current files can be downloaded from www.alliedtelesis.com/support/updates.html.

- 4. Assign an IP address.**

Assign an IP address to the switch interface over which the software files are to be loaded.

5. Load software files onto the switch.

Load the required software and patch onto the switch. See [“Loading Files onto the Switch” on page 5-4 of Chapter 5, Managing Configuration Files and Software Versions.](#)

6. Set the install information.

Set the switch to use the software installed. See [“Install Process” on page 5-13 of Chapter 5, Managing Configuration Files and Software Versions.](#)

7. Reconfigure the switch.

If you have a copy of the recent configuration file stored on your network, you can download it onto the switch too. Otherwise, you must re-enter the configuration.



Caution While flash is compacting, do not restart the switch or use commands that affect the flash file subsystem such as **create**, **edit**, **load**, **rename**, or **delete**. Wait until you get a message that file compaction is complete. Interrupting flash compaction may damages files.

If you accidentally restart the switch, or use commands that affect the subsystem, contact your authorised distributor or reseller. You might have to return the switch to the factory.

Non-Volatile Storage (NVS)

Non-volatile storage (NVS) provides a facility to store information so that it is not destroyed when the switch is reset or powered off. NVS has less capacity than flash memory, but files can be written to NVS more quickly. Files can be copied between NVS and flash. NVS can store module configuration tables, interface configurations, and script files, but not the startup configuration file itself, so the switch cannot boot directly from NVS.

The NVS is organised as blocks of contiguous memory of varying size. A block ID and an index uniquely identifies each block and an owner ID indicates which module created the block. NVS blocks are normally maintained by the modules that created them, but this can also be done manually.

To display information about each block in the NVS including ID, index, owner, size, and creation date, use the [show nv command on page 6-53.](#)

To display the amount of free space in the NVS along with the size of the largest block that can be created, use the [show nv free command on page 6-54.](#)

To delete blocks, use the [delete nv command on page 6-25.](#) Or delete all blocks with the [clear nv totally command on page 6-19.](#)

To display data in the NVS blocks, use the [dump nv command on page 6-26.](#) To change data, use the [modify nv command on page 6-30.](#)

The switch's file subsystem provides a file-based interface to NVS memory that lets you use NVS to store scripts and other files (see [“Working with Files” on page 6-5\).](#)

Secure Digital (SD) Cards

x900-24X switches support Secure Digital cards. These cards are small, removable mass storage devices that expand the amount of memory for storing files. They typically have a much larger capacity than flash or NVS.

You can copy files between the SD card and flash or NVS. You can also use these cards to transfer files, such as configuration files, script files, and release package files from one switch to another or between the switch and a personal computer. Because the switch uses a well-known format on the cards—the file allocation table (FAT) file system—nearly all personal computers understand it.

For more information about the SD card, see the switch's Hardware Reference.

Working with SD cards

File names on an SD card begin with the device name *card*. For more information about naming files, see [“File Naming Conventions” on page 6-3](#).

The switch normally reads and writes files in the root directory of the SD card, and cannot create and manipulate subdirectories on the card. However the switch can read and write files in subdirectories already on the card if you specify an absolute path in the file name. An example of this is `card:/subdir1/subdir2/myscript.scp`.

To display information about the card, use the command:

```
show card
```

To clear the file system on the card, use the command:

```
clear card totally
```

This command may be useful if you have an unreadable card that you want to reformat.

To clear counters and status information on the card, use the command:

```
reset card counters
```

To display counters and status information about the card, use the command:

```
show card counters
```

To test the card slot on front of the switch, use the command:

```
enable test device=card type=slot
```

When the SD card is in use, the SD LED on the front of the switch flashes green.



Caution Do not remove the card from a switch or personal computer when it is in use or you will corrupt data on it.

CompactFlash (CFlash) Cards

x900-48FE and AT-9900 switches support CompactFlash cards. These cards are small, removable mass storage devices that use flash memory. Flash memory is a type of non-volatile, reusable memory that does not require power from a battery to retain stored data. A CompactFlash card can be used to expand the amount of flash memory available for the storage of files on the switch. Files can be manipulated directly by using the command line interface. Any type of data, including releases, patches, GUIs, and configurations can be stored on a CompactFlash card. However, release, patch, and GUI files cannot run from

CompactFlash (CFlash). These files must be loaded into either NVS or flash.

The FAT File System

The file allocation table (FAT) file system is implemented on CompactFlash cards. The FAT file system is a table that provides a map of the clusters that a file is stored in CFlash. Files are stored in one or more clusters that are not necessarily sequential. For every new file, a FAT entry is created that records where each cluster is located and the consecutive order of the clusters. When a file is read the clusters are reassembled using the FAT entry.

There are three variants of the FAT format; these are FAT12, FAT16, and FAT32. The switch implements FAT16.

CompactFlash supports the Allied Telesis long filename implementation. This is not compatible with the FAT16 long filename implementation. For more information about long filenames see [“Working with Files” on page 6-5](#).

Two copies of the file allocation table (FAT) are stored on the CFlash card. These must be kept synchronous and must reflect the current state of CFlash. File allocation tables are updated when a file write is started. This means that there is minimal data loss if the card is removed, or power is lost before a file write completes.

CFlash cards used on the switch must support a hardware access time of no more than 100 nanoseconds. If the CompactFlash cards do not meet the 100 nanosecond requirement they may not work because this is the maximum bus timing allowed.

The switch detects the loss of power during a file write or delete by detecting inconsistency in the FAT tables. The switch recovers the FAT tables and automatically removes the corrupted file.

For more information about the CFlash card, see the switch’s Hardware Reference.

Testing a CFlash card To test CFlash, use the command:

```
enable cflash test start=number end=number
```

To stop the test, use the command:

```
disable cflash test
```

To display progress while testing, use the command:

```
show cflash test
```

You can also use this command when testing is not active to display the blank cluster range on the card.

Working with CFlash cards

Files stored in CFlash are prefixed by “cf.”. When the **load** command is used, the cf. prefix is automatically assigned when the **destination** parameter is set to **cflash**. Files without a device prefix are stored in onboard flash by default. For more information about naming files, see [“File Naming Conventions” on page 6-3](#).

Commands that use a filename as a parameter accept a CFlash file name.

An absolute path specifies the directory structure starting at the root, or start, directory of the CFlash, FAT file system. This is denoted by a slash, either “/”

or “\”, after the CompactFlash device string followed by a slash, for example `dir1/subdir1/`.

A relative path specifies the directory structure starting at the current directory. This is denoted by a directory name after the CompactFlash device, for example `subdir1/subdir2/`. Only the slash “/” is valid.

Release and patch files (with extensions `.rez` or `.paz`) are stored on the switch in a binary format whereas on a PC they are stored in Motorola S-Record format. A CFlash card can contain files in both formats. To handle this situation, files in binary format are copied as normal files via the **copy** command and files in Motorola S-Record format are copied via the **load** command. Using the **load** command performs the task of removing Motorola S-Record information.

To download a patch, release, script, or GUI file stored in CFlash into flash, use the command:

To download a patch, release, or script file stored in CFlash into flash, use the command:

```
load method=cflash destination=flash file=filename
```

For more information on how to load files onto the switch, see “Loading Files onto the Switch” on page 5-4 of Chapter 5, [Managing Configuration Files and Software Versions](#).

Files can be copied to and from CFlash. Where a file already exists with the specified name on the destination device, an error or warning is generated rather than a prompt to overwrite the file. To copy a file to or from CFlash, use the command:

```
copy [device:] filename1.ext [device:] filename2.ext
```

where:

- *device* indicates the physical location of the file, either flash, CFlash, or NVS, with the default being flash
- *filename1* is the name of an existing file
- *filename2* is a valid filename between 1 and 28 characters long that does not already exist

Error messages are displayed when a file write fails. Failure could be due to the card being removed or an error in the card.

When CFlash is in use, the CF LED on the front of the switch is green.



Caution Do not remove the CFlash card when it is in use because you will corrupt data.

To add or set the current directory on the CF card, use the commands:

```
add cflash dir=directory-name
set cflash dir=directory-name
```

To delete a directory from the CF card, use the command:

```
delete cflash dir=directory-name [recurse={true|false}]
```

A directory that contains files or subdirectories is not deleted unless **recurse=true**.

The **set config** command accepts scripts stored on a CF card. When there is a boot.cfg file on the root directory of the card, it runs when there is not one on the onboard flash.

To display information about the CF card, including card size, file count and serial number, enter the command:

```
show cflash
```

Important Anyone with a compliant reader can read CFlash cards so do not keep sensitive data on them.

Command Reference

This section describes the commands available on the switch to support day-to-day operational and management activities.

The shortest valid command is denoted by capital letters in the Syntax section. See “[Conventions](#)” on page xlix of [About this Software Reference](#) for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

activate flash compaction

Syntax ACTivate FLash COMPACTION

Description This command activates the flash compaction process on x900-48FE and AT-9900 switches so you can recover space before the preset threshold is reached that triggers an automatic compaction. The compaction process is usually automatic so manual compaction is not required during normal operations.

Compaction is the process of cleaning up garbage (deleted files) by searching through flash memory, copying valid files to a new block, and erasing old blocks. This operation deletes files in the block being cleared and frees space for new files.

Compaction is necessary because flash memory has a granular erase structure that requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure, the FFS keeps track of the status of each file—whether it is being written, is complete, or is deleted. When the total amount of flash memory for deleted files reaches a preset limit, a compaction process begins.

Compaction may take several seconds when a large amount of flash is involved. However, flash memory operations are unaffected by the process.



Caution While flash is compacting, do not restart the switch or use commands that affect the flash file subsystem such as **create**, **edit**, **load**, **rename**, or **delete**. Wait until you get a message that file compaction is complete. Interrupting flash compaction may damages files.

While compaction is underway, the [show flash command on page 6-50](#) indicates an FFS global operation is compacting. When finished, the global operation shows "none".

Related Commands [show flash](#)

add cflash dir

Syntax ADD CFlash DIR=*directory-name*

where *directory-name* is the name of a directory up to 8 characters long. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

Description This command creates a new directory on the CompactFlash card for x900-48FE and AT-9900 switches.

The **dir** parameter specifies the name of the directory on the CompactFlash card. If the directory specified exists, the directory is not created and a warning message is displayed.

Examples To create a directory named "release", use the command:

```
add cf dir=release
```

Related Commands [delete cflash dir](#)
[set cflash dir](#)
[show cflash](#)

add file

Syntax `ADD File=filename [COMmand=commandstring]
[SCRipt=scriptname] [PERManentredirect] [LIMIT=limit]`

Description This command takes output from a specific command or script and adds it to a text file when you next issue that command or script. This is useful for collecting debug output, and for more information, see [“Sending Output to a Text File” on page 6-8](#). If a file does not exist, one is created. While output is being redirected, the text file cannot be edited, renamed, deleted, or uploaded.

Parameter	Description
File	Name of the text file where you want to send output. One is created if it does not already exist. The <i>filename</i> is in the format <code>[device:]filename.txt</code> and can be: uppercase and lowercase letters digits # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space <i>device</i> indicates the physical location where the file is stored. The default is flash. Default: no default
COMmand	Command whose output is used to generate the text when it is next issued. <i>Commandstring</i> is the command syntax enclosed in quotes. Command and script are mutually exclusive.
SCRipt	Script whose output is used to generate the text when it is next issued. The script is treated as a simple list of commands. Flow control statements are not accepted to ensure that the extra text the script produces is not in the output file. <i>Scriptname</i> has the same format as <i>filename</i> except it must have either a .cfg or .scp extension. Command and script are mutually exclusive.
PERManentredirect	Permanently directs output to the designated text file until the reset file permanentredirect command is issued or the switch is rebooted.
LIMIT	A decimal number from 0 to 1048576 bytes specifying the maximum file size. Default: 204800 bytes

Examples To add output one time only from the **show trace** command to a file called trace.txt command, use the command:

```
add fi=trace.txt com="show trace"
```

To permanently add output from the **show debug** command to a file called debug2.txt command, use the command:

```
add fi=debug2.txt com="show debug"
```

Related Commands [create file](#)
[reset file permanentredirect](#)
[show file permanentredirect](#)

clear card totally

Syntax CLeAr CArd TOTAlly

Description This command completely clears the file system on an SD card for x900-24X switches, and requires a user with Security Officer privilege when the switch is in security mode.

Clearing a memory card repartitions and reformats it. If you have an unreadable card, you can use this command to reformat it. However, clearing a card is not required for normal operations, and this command is intended primarily for troubleshooting. Clearing an SD card may take several seconds and the CLI does not respond while it is in progress.



Caution This command destroys all files on the card. Files cannot be salvaged after a card has been cleared.

Related Commands

[clear flash totally](#)
[clear nvs totally](#)
[show card](#)

clear cflash

Syntax CLeAr CFlash

Description This command deletes the contents of the CompactFlash card for x900-48FE and AT-9900 switches. Entries in the file allocation table and the root directory are cleared.

Do not use this command to destroy confidential data because complete data erasure is not guaranteed. All data remains on the CFlash card but is inaccessible through conventional means.

Examples To delete contents from the CFlash card, use the command:

```
cl cf
```

Related Commands

[show cflash](#)

clear flash totally

Syntax CLeAr FLash TOTAlly

Description This command completely clears the file system in flash memory by erasing and reformatting it. Clearing flash is not required for normal operations. This command is intended for troubleshooting, and requires a user with Security Officer privilege when the switch is in security mode.

Clearing flash may take several minutes on x900-24X switches, and the CLI does not respond while it is in progress.

Erasing flash may take several minutes on x900-48FE and AT-9900 switches. While it is underway, the [show flash command on page 6-50](#) shows that the FFS global operation is “erasing”. When the operations finishes, “Erasure is successfully completed” is displayed and the global operation shows “none”.



Caution This command destroys all files in flash, including essential ones with product software, licence information, and install information. Files cannot be salvaged after flash has been cleared.

Related Commands [delete file](#)
[clear nvs totally](#)

clear nvs totally

Syntax CLeAr NVS TOTAlly

Description This command completely clears the file system in non-volatile storage (NVS) memory by erasing and reformatting it. Clearing NVS is not required for normal operations. This command is intended for troubleshooting, and requires a user with Security Officer privilege when the switch is in security mode.



Caution This command destroys all files in NVS. You cannot salvage files after you clear NVS.

Related Commands [delete file](#)
[modify nvs](#)
[show nvs](#)
[show nvs free](#)
[clear flash totally](#)

copy

Syntax for x900-24X `COPY src-filename.ext dest-filename.ext`

where:

- *src-filename* is the name of an existing file in the format [*device*:]*filename.ext*. Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ! # \$ % & ' () + , - . ; = @ [] ^ _ ` { } ~ and space

device indicates the physical location where each file is stored, either flash, NVS, or memory card. The default is flash.

.ext is an extension, such as .txt or .cfg. The original file and the copy must have the same extensions.

- *dest-filename* is name of a destination file in the same format as *src-filename*. The filename must not already exist.

Wildcard characters * and | may be used in the file name to specify a group of files. For more information about wildcards, refer to [“Using Wildcards” on page 6-7](#).

Syntax for x900-48FE and AT-9900 `COPY [device:]src-filename.ext [device:]dest-filename.ext`

When stacking is enabled `COPY [source-hostid:][device:]filename1.ext
[dest-hostid:][device:]filename2.ext`

where:

- *device* indicates the physical location of the file, and is either flash, CFlash, or NVS. The default is flash.
- *src-filename* is the source file and name of an existing file in the format [*device*:]*filename.ext*. Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ~ ' ! @ # \$ % ^ & () _ - { }

.ext is an extension, such as .txt or .cfg. The original file and the copy must have the same extensions.

- *dest-filename* is name of a destination file in the same format as *src-filename*. The filename must not already exist.
- *source-hostid* indicates the host ID of the stack member that holds the file to be copied. This variable is optional but if present, the *dest-hostid* is required; if not present, the **copy** command functions in the standard way.
- *dest-hostid* indicates stack members to receive the copied file:
 - a unique number from 1 to 32
 - a range of unique numbers from 1 to 32
 - a comma-separated list

This variable is optional but if present, the *source-hostid* is required; if not present, the **copy** command functions in the standard way.

Description This command copies files to another location. For x90048FE and AT-9900 switches, it copies a text file from one type of memory to another, either flash, CFlash, or NVS.

Copying an x900-24X package file to flash may take several minutes. If you are copying a file between stack members on x900-24X switches, the maximum allowable source file size is 500kBytes.

Example To copy the file `admin.cfg` to the file `admin2.cfg` on an x900-24X switch, use the command:

```
cop admin.cfg admin2.cfg
```

To copy all script files from an SD card to flash on an x900-24X switch, use the command:

```
cop card:*.scp flash:
```

To copy the file `admin.cfg` on NVS to the file `admin2.cfg` on flash, use the command:

```
cop nvs:admin.cfg admin2.cfg
```

To copy the file `config.cfg` from flash to CFlash on an AT-9900 switch, use the command:

```
cop config.cfg cf:config.cfg
```

To copy `file1.txt` on an x900FE switch from a stack member with host ID 1 to stack members with host IDs 2 through 4, and also 6, and rename it "file2.txt", use the command:

```
cop 1:file1.txt 2-4,6:file2.txt
```

To copy `file1.txt` on an AT-9900 switch from the stack member with host ID 1 to the stack member with host ID 2 without renaming it, use the command:

```
cop 1:file.txt 2:file.txt
```

While connected to the stack member with host ID 1 on an x900-48FE switch, use a host-directed command to direct host ID 2 to copy `file.txt` and rename it "file2.txt":

```
2:cop file.txt file2.txt
```

Note that you **cannot** use a host-directed command to direct the **copy** command to more than one stack member at a time. For example, "1-3: `cop file.txt file2.txt`" returns an error.

Related Commands [delete file](#)
[rename](#)
[show file](#)

create file

Syntax CREate File=*filename* [FORCE] [COMmand=*commandstring*]
[SCRipt=*scriptname*] [PERManentredirect] [LIMIT=*limit*]

Description This command creates a text file containing output from a specific command or script. This is useful for collecting debug output, and for more information, see [“Sending Output to a Text File” on page 6-8](#). The file cannot be edited, renamed, deleted, or uploaded while it is receiving input.

Parameter	Description
File	Name of the text file that you want to create. The <i>filename</i> is in the format [<i>device</i>]: <i>filename</i> .txt and can be: uppercase and lowercase letters digits # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space <i>device</i> indicates the physical location where the file is stored. The default is flash. Default: no default
FORCE	Overwrites the text file if one already exists. If force is not specified and the file exists, the command has no effect.
COMmand	Command whose output is used to generate the text when it is next issued. <i>Commandstring</i> is the command syntax enclosed in quotes. Command and script are mutually exclusive.
SCRipt	Script whose output is used to generate the text when it is next issued. The script is treated as a simple list of commands. Flow control statements are not accepted to ensure that the extra text the script produces is not in the output file. <i>Scriptname</i> has the same format as <i>filename</i> except it must have either a .cfg or .scp extension. Command and script are mutually exclusive.
PERManentredirect	Permanently directs output to the designated text file until the reset file permanentredirect command is issued or the switch is rebooted.
LIMIT	A decimal number from 0 to 1 048 576 bytes specifying the maximum file size. Default: 204 800 bytes

Examples To permanently direct all debug output from the BGP module to a file named bgp.txt, use the command:

```
cre fi=bgp.txt com="enable bgp debug=all" perm
```

Related Commands [add file](#)
[reset file permanentredirect](#)
[show file permanentredirect](#)

delete cflash dir

Syntax `DELEte CFlash DIR=directory-name [RECURSE={True|False}]`

where *directory-name* is the name of a directory, up to eight characters long. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

Description This command removes a directory from the CompactFlash card for x900-48FE and AT-9900 switches. When the directory contains files or subdirectories, it is not deleted.

The **dir** parameter specifies the name of the directory on the CompactFlash card that is to be deleted.

The **recurse** parameter specifies whether files and subdirectories within the directory are to be deleted. The default is **false**.

Examples To delete the directory called "release", use the command:

```
del cf dir=release
```

Related Commands [add cflash dir](#)
[set cflash dir](#)
[show cflash](#)

delete file

Syntax for x900-24X `DELEte File=filename`

where *filename* is the name of an existing file or group of files in the format [*device*:]*filename.ext*. Valid characters are:

- uppercase and lowercase letters
- digits
- # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space

device indicates the physical location where the file is stored, either flash, NVS, or memory card. The default is flash.

.ext is an extension such as .txt or .cfg.

Wildcard characters * and | may be used in the file name to specify a group of files. For more information about wildcards, refer to [“Using Wildcards” on page 6-7](#).

Syntax for x900-48FE and AT-9900

```
DELeTe File=filename
```

where *filename* is a file identifier in the format [*device:*]filename.ext. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

device specifies the physical location where the file is stored, which is flash.

If a colon is in the file name, it is assumed that it includes a device name.

For information about wildcards, refer to [“Using Wildcards” on page 6-7](#).

Description

This command deletes specific files. It requires a user with security officer privilege when the switch is in security mode.

The GUI resource file that the AT-9900 is set to use can be deleted when the GUI is disabled. GUI resource files have an .rsc extension. Use the **show install** command and check the Current Install section in the output to see which resource file is currently set. See the [disable gui command on page 3-19 of Chapter 3, Using the Graphical User Interface \(GUI\) on AT-9900 Series Switches](#) for more information about disabling the GUI.

Note that you cannot delete the preferred software release or the current boot configuration file with this command. If you want to delete the files without specifying new preferred files, first use the **delete install=pref** command or **set config=none** to stop the files from being preferred.



Caution Files that contain patches, product software, licences, and configurations are vital to the operation of the switch and should be deleted only after careful consideration.

Examples

To delete all the patch files on x900-48FE and AT-9900 switches, use the command:

```
delete file=*.pat
```

To delete the file startup1.cfg, use the command:

```
del startup1.cfg
```

To delete all coredump files in flash with names that begin with the letter s on an x900-24X switch, use the command:

```
del fi=s*.core
```

To delete all script files on all storage devices, use the command:

```
del fi=*.scp
```

Related Commands

[copy](#)
[rename](#)
[show file](#)

delete nvs

Syntax `DELEte NVS Block=id INDEx=index`

where:

- *id* is the block identifier in hexadecimal.
- *index* is the block index in hexadecimal.

Description This command deletes a block from non-volatile storage (NVS) on x900-48FE and AT-9900 switches. It requires a user with Security Officer privilege when the switch is in security mode. The block must be identified by **block** and **index**.

Examples To delete the NVS block with a block id and index of 99, use the command:

```
del nvs b=99 index=99
```

Related Commands [clear nvs totally](#)
[dump nvs](#)
[modify nvs](#)
[show nvs](#)
[show nvs free](#)

disable cflash test

Syntax `DISAbLe CFlash TEST`

Description This command stops the test on the CompactFlash card on x900-48FE and AT-9900 switches.

Related Commands [enable cflash test](#)
[show cflash](#)

dump nvs

Syntax DUMP NVS [BLOCK=*id*] [INDEX=*index*] [LENGTH=*length*]
[OFFSET=*offset*] [SIZE={BYTE | LONG | WORD}]

where:

- *id* is the block ID in hexadecimal.
- *index* is the block index in hexadecimal.
- *length* is the length of data to be dumped in hexadecimal.
- *offset* is the offset into the data to start dumping from in hexadecimal.

Description This command dumps data from a non-volatile storage (NVS) block on x900-48FE and AT-9900 switches (Figure 6-1, Table 6-1). The **size** parameter specifies whether the data should be displayed grouped as bytes, longwords, or words.

The **block**, **index**, **length**, **offset**, and **size** parameters are required the first time the command is used after a reboot; thereafter, they are optional. If not specified, values are used from the previous invocation.

If **offset** is not specified, the dump continues from the end of the previous display. If **offset** is specified without a value, the value from the previous invocation is used.

Figure 6-1: Example output from the **dump nvs** command

ID: 32	Index : 03	Offset: 00000000	Length: 00000050	Size: LONG
Offset	Data	ASCII		
00000000	00010040 00000000 00000000 00000000	...@.....		
00000010	00010000 00000004 00000007 00000000		
00000020	00000000 00000000 00000000 00000000		
00000030	00000000 00000000 00000000 00000000		
00000040	00000000 00000000 00000000 00000000		

Table 6-1: Parameters in output of the **dump nvs** command

Parameter	Meaning
ID	Block ID (in hexadecimal) of the block displayed.
Index	Block index in (hexadecimal) of the block displayed.
Offset	Offset (in hexadecimal) of the data displayed.
Length	Length of data in (hexadecimal) displayed.
Size	Whether the units of data are displayed as bytes, longwords, or words.
Offset	Offset of the current record from the ID, Index and Offset specified in the header.
Data	The data.
ASCII	ASCII representation of the data.

Related Commands

- [clear nvs totally](#)
- [delete nvs](#)
- [modify nvs](#)
- [show nvs](#)
- [show nvs free](#)

edit

Syntax for x900-24X `EDit [filename]`

where *filename* is in the format [*device*:]filename.ext. Valid characters are:

- uppercase and lowercase letters
- digits
- # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space

device indicates the physical location where the file is stored, either flash, NVS, or memory card. The default is flash.

.ext is a text file extension, such as .txt or .cfg.

Syntax for x900-48FE and AT-9900 `EDit [filename]`

where *filename* is in the format [*device*:]filename.ext. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

device specifies the physical location where the file is stored, either CFlash, NVS, or flash. The default is flash. If a colon is in the filename, it is assumed that it includes a device name.

.ext is a 3-letter text file extension, such as .cfg and .txt.

Description This command invokes the switch's built-in full-screen text editor to edit a text file. This command requires a user with Security Officer privilege when the switch is in security mode.

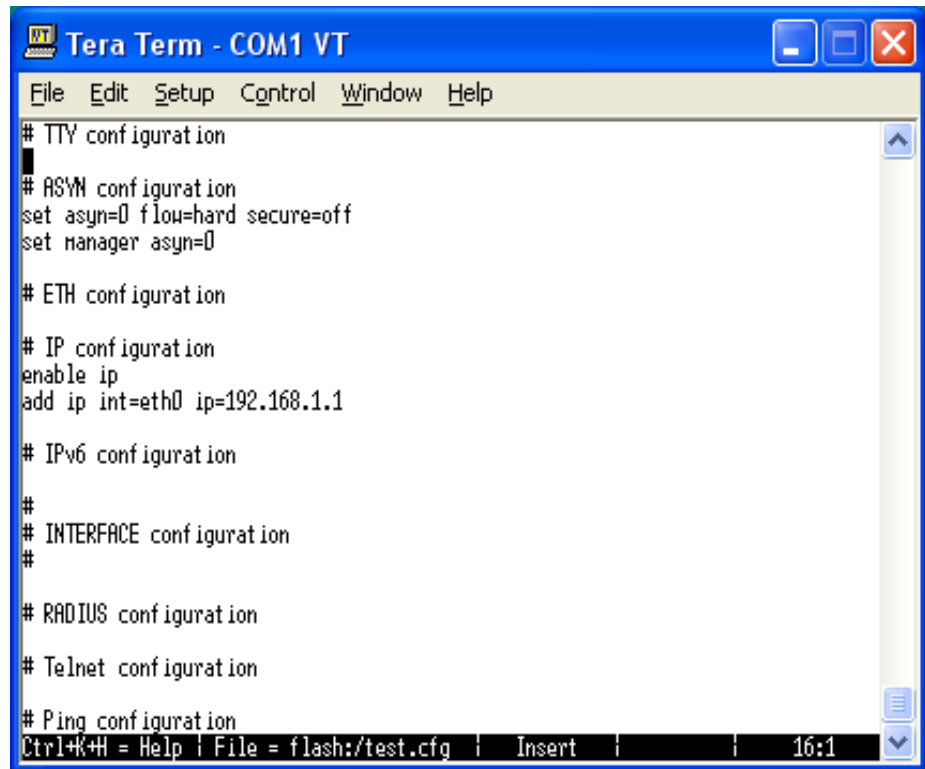
If a filename is specified and it already exists, then the editor loads it on the system. If no filename is specified, the editor prompts you for one when you exit it.

Before starting the editor make sure your terminal, terminal emulation program, or Telnet client is 100% compatible with a VT100 terminal. The editor uses VT100 control sequences to display text on the terminal, and accepts command sequences in the following table.

Cursor Movement		Delete	
or Ctrl+Z	Up one line	Ctrl+T	Delete word right
or Ctrl+X	Down one line	Ctrl+Y	Delete line
	Right one character		
	Left one character	Block Operations	
Ctrl+B	Start of file	Ctrl+K,B	Begin block mark
Ctrl+D	End of file	Ctrl+K,D	Unmark block
Ctrl+A	Start of line	Ctrl+K,U	Cut block to buffer
Ctrl+E	End of line	Ctrl+K,C	Copy block to buffer
Ctrl+U	Up one screen	Ctrl+K,V	Paste block from buffer
Ctrl+V	Down one screen	Ctrl+K,Y	Delete block
Ctrl+F	Word right		
Search		Exit	
Ctrl+K,F	Find text	Ctrl+K,X	Exit editor; save file
Ctrl+L	Repeat last find	Ctrl+C	Quit editor; do not save file
Miscellaneous			
Ctrl+I	Insert mode	Ctrl+O	Overstrike mode
Ctrl+W	Refresh the screen	Ctrl+K,H	Display help screen
Ctrl+K,O	Open a file		

Figure 6-2 shows an example of the text editor screen. The VT100 screen supports 24 lines, unlike a PC. Lines 1–23 display the text of the file being edited; line 24 at the bottom of the screen is the status bar and command line. The status bar displays the current filename, line and column position in the file, and the editing mode (overstrike or insert). When additional command information is required such as a filename or search text, a prompt is displayed in the status bar.

Figure 6-2: The editor screen layout.



The editor edits one file at a time but you can transfer text between files by using the cut and paste operations.

Obtain help at any time while in the editor by pressing [Ctrl+K,H]; that is, holding down the Ctrl key, pressing K, then the H key.

Examples To edit a file in flash named show.scp, use the command:

```
ed show.scp
```

To edit a file in NVS called setup1.cfg, use the command:

```
ed nvs:setup1.cfg
```

Related Commands [delete file](#)
[show file](#)

enable cflash test

Syntax ENABle CFlash TEST START=*number* END=*number*

where *number* is a positive integer. The upper limit is dependent on the CompactFlash card used.

Description This command tests the CompactFlash card on x900-48FE and AT-9900 switches by writing a pattern to a cluster and then reading it back. The test fails and an error message is displayed when the cluster referred to is in the file allocation table.

The **start** parameter specifies what cluster to start the test at.

The **end** parameter specifies what cluster to finish the test at. The **end** parameter must be greater than the **start** parameter.

Related Commands [disable cflash test](#)
[show cflash test](#)

modify nvs

Syntax MODify NVS Block=*id* INDeX=*index* OffSet=*offset*
Size={Byte|Long|Word} Value=*value-list*

where:

- *id* is a block ID number in hexadecimal.
- *index* is a block index number in hexadecimal.
- *offset* is the offset in hexadecimal within the block where the values should be written.
- *value-list* is a list of values in hexadecimal, separated by commas.

Description This command allows the contents of a non-volatile storage (NVS) block to be modified on x900-48FE and AT-9900 switches. The block must be identified by **block** and **index**.

The command contiguously writes data values into the block starting at the specified **offset** and padded to **size** (length). None of the data values can require more space than the specified **size**.

Examples To set the third byte of the NVS block with a block ID and index of 99, to the value 99, use the command:

```
mod nvs b=99 ind=99 o=3 s=b v=fe
```

Related Commands [clear nvs totally](#)
[delete nvs](#)
[dump nvs](#)
[show nvs](#)
[show nvs free](#)

purge file translationtable

Syntax PURge File Translationtable={ALl|UPdate}

Description This command resynchronises the translation table to file contents in memory on x900-48FE and AT-9900 switches. It is possible that the translation table and memory could become unsynchronised, for example in the event of a power outage during a file creation or deletion operation. This might result in files in memory that are not represented in the translation table, and visa versa.

To use this command when CompactFlash is activated, the current working directory must be the root directory.

The **all** option completely rebuilds the translation table. All long filenames are lost. The table continues to support subsequent long filename creation and management.

The **update** option restores all valid long filenames to the appropriate table entries after the table has been rebuilt. Long filenames that are not reconciled to the new table and table entries that are not confirmed to be in memory are deleted. This leaves a translation table that has maintained all of its previously valid data, and disposed of the rest. The table continues to support all subsequent long filename creation and management.

Examples To rebuild the translation table and remove all long filenames, use the command:

```
pur fi t=al
```

To rebuild the translation table and attempt to recover all long filename data, use the command:

```
pur fi t=up
```

Related Commands [show file](#)

rename

Syntax `REName src-filename dest-filename`

Description This command renames files and requires a user with security officer privilege when the switch is in security mode. Please note that you cannot rename the preferred software release or the current boot configuration file.



Caution Files that contain product software, patches, licences, and configurations are vital to the operation of the switch and should be renamed only after careful consideration.

Parameter	Description
<i>src-filename</i>	<p>Filename of an existing source file. When the source is a secure file type, such as LIC or RND, its filename extension must match that of destination file.</p> <p>Default: no default</p> <hr/> <p>With the following:</p> <ul style="list-style-type: none"> in [device:] filename..ext format. When no device is specified, flash is assumed. Source and destination devices must be the same. uppercase and lowercase letters digits for x900-48FE and AT-9900 switches ~ ' ! @ # \$ % ^ & () _ - { } for x900-24X switches ! # \$ % & ' () + , - . ; = @ [] ^ _ ` { } ~ and space
<i>dest-filename</i>	<p>Filename of the destination file, which must not already exist.</p> <p>Default: no default</p> <hr/> <p>With the following:</p> <ul style="list-style-type: none"> in [device:] filename .ext format. When no device is specified, flash is assumed. Source and destination devices must be the same. uppercase and lowercase letters digits for x900-48FE and AT-9900 switches ~ ' ! @ # \$ % ^ & () _ - { } for x900-24X switches ! # \$ % & ' () + , - . ; = @ [] ^ _ ` { } ~ and space

Examples To rename boot.cfg to saveboot.cfg, use the command:

```
ren boot.cfg saveboot.cfg
```

Related Commands

- [copy](#)
- [delete file](#)
- [show file](#)

reset card counters

Syntax RESET CARd COUnters

Description This command clears counters and status information about Secure Digital (SD) memory cards on x900-24X switches.

Related Commands [show card counters](#)

reset file permanentredirect

Syntax RESET File[=*filename*] PERManentredirect

Description This command closes one or all text files so that they no longer receive input from commands or scripts. After the file closes, it can be uploaded or edited

Parameter	Description
File	Name of the text file to close. If no file is specified, all text files are closed. The <i>filename</i> is in the format [<i>device</i> :]filename.txt and can be: uppercase and lowercase letters digits # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space <i>device</i> indicates the physical location where the file is stored. The default is flash. Default: no default

Examples To reset the bgp.txt file so that it no longer receives output from the **enable bgp debug=all** command (previously set), use the command:

```
reset fi=bgp.txt perm
```

Related Commands [add file](#)
 [create file](#)
 [show file permanentredirect](#)

set cflash dir

Syntax SET CFlash DIR=*directory-name*

where *directory-name* is the name of a directory, up to eight characters long. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

Directories can be separated by a forward or back slash.

Description This command sets the current directory on the CompactFlash card on x900-48FE and AT-9900 switches.

The **dir** parameter specifies the name of the directory that is to be set as the current directory. The directory must exist. Files referred to by "cf:name.ext" are looked for in the current directory. To set the directory to the base directory, specify "/" as the value for the **dir** parameter.

Examples To set the current CompactFlash directory to /scripts/switch, use the command:

```
set cf dir=/scripts/switch
```

Related Commands [add cflash dir](#)
[delete cflash dir](#)
[show cflash](#)

show card

Syntax SHow CArd

Description This command displays information about the file system on a Secure Digital (SD) memory card for x900-24X switches (Figure 6-3, Table 6-2).

Figure 6-3: Example output from the **show card** command

```
Volume label ..... SD_512
Files total size ..... 13367336 bytes (30 files)
Used space ..... 13434880 bytes
Free space ..... 497754112 bytes
Total space ..... 511188992 bytes
```

Table 6-2: Parameters in output of the **show card** command

Parameter	Meaning
Volume label	Volume label of the card when a label is present.
Files total size	Sum of the sizes of all files in the root directory of the SD card. The number of files in the root directory is in parentheses.
Used space	Space that all files, subdirectories, and formatting overhead use on the card.
Free space	Available space on the card.
Total space	Total capacity of the file system on the card.

Related Commands [show card counters](#)
[show file](#)
[show flash](#)
[show nvs](#)

show card counters

Syntax SHow CARd COUnters

Description This command displays counters and status information about data transfers to and from memory cards on x900-24X switches (Figure 6-4, Table 6-3).

Figure 6-4: Example output from the **show card counters** command

Card counters	
deviceInitialised	1
dataReads	13813
dataBlocksRead	27340
dataBytesRead	13998080
operationsFailed	0
deviceFormatted	0
dataWrites	1578
dataBlocksWritten	3111
dataBytesWritten	1592832
dataWritesProhibited	0
Card diagnostic counters	
cardInitRetried	0
sectorReadRetried	0
sectorWriteRetried	0
commandChecksumErrors	0
readChecksumErrors	0
writeChecksumErrors	0
initialiseTimeouts	0
cardInitFailed	0
sectorReadFailed	0
sectorWriteFailed	0
cardFormatFailed	0
commandOtherErrors	0
readOtherErrors	0
writeOtherErrors	0
busErrors	0
Card command results	
Last command:	
command	17
commandResponse	00h
dataReadResponse	FEh
dataWriteResponse	-
Last failed command:	
command	-
commandResponse	-
dataReadResponse	-
dataWriteResponse	-

Table 6-3: Parameters in output of the **show card counters** command

Parameter	Meaning
Card counters—for normal operations. These apply to any type of memory card.	
deviceInitialised	Number of times the switch identified and initialised a memory card. The first time the switch uses a card after it is inserted into the card slot, it resets the card, reads its type and size, and mounts its file system. Data on the card is not altered.
deviceFormatted	Number of times the switch formatted the memory card with the clear card totally command.
dataReads	Number of data read calls made to the memory card device driver.
dataWrites	Number of data write calls made to the memory card device driver.
dataBlocksRead	Number of blocks of file system data read from the memory card. Each block contains one FAT sector of 512 bytes.
dataBlocksWritten	Number of blocks of file system data written to the memory card. Each block contains one FAT sector of 512 bytes.
dataBytesRead	Number of bytes of file system data read from the memory card.

Table 6-3: Parameters in output of the **show card counters** command (cont)

Parameter	Meaning
dataBytesWritten	Number of bytes of file system data written to the memory card.
operationsFailed	Number of card operations, such as initialise, format, data read, or data write, that have failed.
dataWritesProhibited	Number of data write operations that failed because the memory card is write-protected.
Card diagnostic counters—for failed operations. These apply only to Secure Digital cards.	
cardInitRetried	Number of times a card initialise was retried because of an error.
cardInitFailed	Number of times a card initialise failed despite retries.
sectorReadRetried	Number of times a data block read was retried because of an error.
sectorReadFailed	Number of times a data block write was retried because of an error.
sectorWriteFailed	Number of times a data block write failed despite retries.
cardFormatFailed	Number of times a card format operation failed despite retries of its constituent operations.
commandChecksumErrors	Number of times the card rejected a command because of a cyclic redundancy checksum (CRC) error.
commandOtherErrors	Number of times the card reported a command error other than a CRC error.
readChecksumErrors	Number of times the switch detected a CRC error in data read from the card.
readOtherErrors	Number of times the switch detected or the card reported a data read error other than a CRC error.
writeChecksumErrors	Number of times the card reported a CRC error in data written to the card.
writeOtherErrors	Number of times the card reported a data write error other than a CRC error.
initialiseTimeouts	Number of times the card was not ready after initialisation.
busErrors	Number of bus errors the card bus driver reported.
Card command results—command codes the switch sends to the SD card and the corresponding response codes the card returns to the switch. These codes are part of the physical layer bus protocol for SD cards defined by the SD Card Association.	
Last command	Command code and responses for the most recent command the switch sent to the SD card.
command	Command code (a decimal number) sent to the card. Shown as a dash when no commands have been sent to the card since the card counters were last reset.
commandResponse	Command response code (a hexadecimal number) the card received in response to each command. 00h indicates success; any other value indicates a command error. Shown as a dash when no command responses have been received since the card counters were last reset.

Table 6-3: Parameters in output of the **show card counters** command (cont)

Parameter	Meaning
dataReadResponse	Data read response code (a hexadecimal number) the card received in response to commands that read file system data from the card. 00h indicates success; any other value indicates a data read error. Shown as a dash when no data read responses have been received since the card counters were last reset.
dataWriteResponse	Data write response code (a hexadecimal number) the card received in response to commands that write file system data to the card. 00h indicates success; any other value indicates a data write error. Shown as a dash when no data write responses have been received since the card counters were last reset.
Last failed command	Command code and responses for the most recent failed command the switch sent to the SD card. The switch treats a command as failed when the command response, data read response, or data write response indicate an error.
command	Command code (a decimal number) sent to the card. Shown as a dash if no commands have failed since the card counters were last reset.
commandResponse	Command response code from the card (a hexadecimal number). 00h indicates success; any other value indicates a command error. Shown as a dash when no command responses from failed commands have been received since the card counters were last reset.
dataReadResponse	Data read response from the card (a hexadecimal number). 00h indicates success; any other value indicates a data read error. Shown as a dash if no data read responses from failed commands have been received since the card counters were last reset.
dataWriteResponse	Data write response from the card (a hexadecimal number). 00h indicates success; any other value indicates a data write error. Shown as a dash if no data write responses from failed commands have been received since the card counters were last reset.

Related Commands [reset card counters](#)
[show card](#)

show cflash

Syntax SHow CFlash

Description This command displays information about the CompactFlash card on x900-48FE and AT-9900 switches (Figure 6-5, Table 6-4).

Figure 6-5: Example output from the **show cflash** command

```
Compact Flash
-----
Current Directory: \
    Number of files ..... 2
    Number of directories ..... 1
    Bytes used ..... 4957

Card Information:
    Hardware detected ..... Yes
    Serial Number..... SST41602120000003672
    Size ..... 125056 KB
    Used ..... 4 KB ( 9 files)
    Free ..... 125052 KB
    Global State ..... None
-----
```

Table 6-4: Parameters in output of the **show cflash** command

Parameter	Meaning
Number of files	Number of files in the current directory.
Number of directories	Number of subdirectories in the current directory.
Bytes used	Space taken up by the files in the current directory.
Current Directory	Current directory as set by the set cflash dir command.
Serial Number	Serial number read off the card.
Size	Size in kilobytes of the CompactFlash card.
Used	Kilobytes of space already used on the CompactFlash card.
Free	Kilobytes of free space on the CompactFlash card.
Global State	Current global operation, if any. Possible values are None, Reading, Writing, Loading, Testing, or Erasing.

Example To display information about the CompactFlash card, use the command:

```
sh cf
```

Related Commands

- [add cflash dir](#)
- [set cflash dir](#)
- [show cflash counters](#)
- [show cflash file](#)
- [show cflash test](#)

show cflash counters

Syntax SHow CFlash COunters

Description This command displays all the CompactFlash counters concerning sectors read and write (Figure 6-6, Table 6-5). Valid for x900-48FE and AT-9900 switches only.

Figure 6-6: Example output from the **show cflash counters** command

```
Compact Flash Counters

Successful operations ..... 10
Read Errors ..... 0
Write Errors ..... 0
Bad Clusters ..... 0
Incomplete Clusters ..... 0
Hardware Errors ..... 0
FAT entries not found ..... 0
Sectors out of range ..... 0
Operations retried ..... 0
```

Table 6-5: Parameters in output of the **show cflash counters** command

Parameter	Meaning
Successful operations	Number of operations carried out successfully.
Read Errors	Number of errors encountered trying to read the card.
Write Errors	Number of errors encountered trying to write the card.
Bad Clusters	Number of bad clusters encountered.
Incomplete Clusters	Number of incomplete cluster chains encountered.
Hardware Errors	Number of errors reported by the compact flash card.
FAT entries not found	Number of times an error has occurred because there was not entry in the file allocation table.
Sectors out of range	Number of sectors that have been accessed that are outside of the valid range.
Operations retried	Number of operations that have been retried.

Examples To display information about CompactFlash counters, use the command:

```
sh cf cou
```

Related Commands [show cflash](#)
[show cflash file](#)
[show cflash test](#)

show cflash file

Syntax SHow CFlash FIle

Description This command displays the files in the current directory on x900-48FE and AT-9900 switches (Figure 6-7, Table 6-6).

Figure 6-7: Example output from the **show cflash file** command

Name	Attr	Size	Cluster	Sector	Created
RELEASE	----d--	0	2	527	15-Jul-2002:07:35:16.0
CONFIG	----d--	0	210	1359	15-Jul-2002:09:33:02.0
DEFAULT CFG	-----a-	2451	618	2991	15-Jul-2002:10:45:54.100
FILE	TEXT -----a-	59	617	2987	15-Jul-2002:10:45:36.98

Table 6-6: Example output from the **show cflash file** command

Parameter	Meaning
Name	Name of the file on the CompactFlash card. This is displayed as it is stored i.e. in capitals with the last three characters making up the extension.
Attr	File attributes as stored on the CompactFlash card. Valid attributes are: d (directory), a (archive), h (hidden), s (system). Apart from the directory attribute, the switch ignores the attributes.
Size	Size of the file in bytes, as a decimal number.
Cluster	Cluster that marks the start of the files data.
Sector	First sector of the cluster.
Created	Date and time the file was created.

Related Commands [show cflash](#)
 [show cflash counters](#)
 [show cflash test](#)

show cflash test

Syntax SHow CFlash TEST

Description This command displays the testing process on the CompactFlash card on x900-48FE and AT-9900 switches when the **enable cflash test** command is active and testing is still in process. If no testing is running, this command displays the blank cluster range on the CFlash card. Types of output are:

- when testing is running (Figure 6-8, Table 6-7)
- when testing is not running (Figure 6-9, Table 6-8 on page 6-43)

Figure 6-8: Example output from the **show cflash test** command when testing

```
Test Progress
  Starting cluster      = 700
  Ending cluster       = 1700
  Current cluster      = 1185

  Passed clusters number = 485
  Passed sectors number  = 1940

  Failed clusters number = 0
  Failed sectors number  = 0
  Used Clusters encountered = 0

  Duration..... 4417 ms
```

Table 6-7: Parameters in output of the **show cflash test** command when testing

Parameter	Meaning
Starting cluster	Cluster that was entered as the start parameter for the enable cflash test command.
Ending cluster	Cluster that was entered as the end parameter for the enable cflash test command.
Current cluster	Cluster that is currently being tested.
Passed clusters number	Number of cluster that have passed the test.
Passed sectors number	Number of sectors that have passed the test.
Failed clusters number	Number of clusters that have failed the test.
Failed sectors number	Number of sectors that have failed the test.
Used Clusters encountered	Number of used clusters encountered.
Duration	Time taken to perform the test.

Figure 6-9: Example output from the **show cflash test** command when not testing

```
Clusters available for testing
  Ranges:
    [3--3] [496--46846]
  Number of free clusters = 46352
  Number of ranges       = 2
  Number of used clusters = 495
```

Table 6-8: Parameters in output of the **show cflash test** command when not testing

Parameter	Meaning
Ranges	Ranges of free sectors on the CompactFlash card.
Number of free clusters	Number of clusters marked as free in the file allocation table.
Number of ranges	Number of ranges detected.
Number of used clusters	Number of clusters marked as used.

Related Commands

[disable cflash test](#)
[enable cflash test](#)
[show cflash](#)
[show cflash counters](#)
[show cflash file](#)

show ffile

Syntax `SHoW FFile[=file-identifier] [CHECK]`

where *file-identifier* is a valid FFS file identifier in the format module\filename.ext. Wildcards are allowed in any of the elements. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

Description This command displays a list of the files in the Flash File System (FFS) on x900-48FE and AT-9900 switches that match the specified file identifier (Figure 6-10, Table 6-9). If a file identifier is not specified, all files are displayed. Wildcards can be used to replace any part of the file identifier to allow a more selective display.

The **check** parameter specifies that the file data checksums are to be verified. Output with this parameter may require a few more seconds for larger files.

Figure 6-10: Example output from the **show ffile** command

module	name	type	size	file date & time	address	check
	ops	cfg	2610	18-Feb-2003 03:50:12	FECD734C	-
	help	hlp	94790	21-Jan-2003 07:57:41	FECC005C	-
	config	ins	32	03-Mar-2003 10:24:43	FEB05DC0	-
	prefer	ins	64	28-Feb-2003 06:08:59	FEADD1B4	-
	longname	lfn	60	18-Feb-2003 03:54:54	FECD7E60	-
	feature	lic	39	21-Jan-2003 07:57:59	FECD72E4	-
	random	rnd	3904	03-Mar-2003 10:44:43	FEB05E20	-
	d_410e00	rsc	2449712	19-Feb-2003 09:09:09	FECD7F5C	-
inst	release	lic	96	18-Feb-2003 03:54:09	FECD7DC0	-
load	melistst	paz	6108	03-Mar-2003 10:24:09	FEB045A4	-
load	52-251	rez	2795756	28-Feb-2003 05:59:36	FE82F12C	-

flash use:						
	files	5354100 bytes (12 files)				
	garbage	178988 bytes				
	free	1675872 bytes				
	block size .	131072 bytes				
	total	7340032 bytes				

Table 6-9: Parameters in output of the **show ffile** command

Parameter	Meaning
module	Module that created the file.
name	filename.
type	File type.
size	Size of the file in bytes shown as a decimal number.
file date & time	Date and time the file was created.
address	Base address of the file in hexadecimal.

Table 6-9: Parameters in output of the **show ffile** command (cont)

Parameter	Meaning
check	Result of the file data check (if check was specified).
files	Number of bytes of flash memory used by valid files.
garbage	Number of bytes of flash memory used by deleted files.
free	Number of bytes of flash memory free.
total	Total size of flash memory.

Examples To display all the release files created by the Loader module, use the command:

```
sh ff=load\*.rez
```

Related Commands [show cflash file](#)
[show nvs](#)
[show file](#)

show file

Syntax for x900-24X `SHoW FiLe[=filename] [DEvice={ALl|FLash|NVs|CArD}]`

where *filename* is in the format [*device*:]*filename*.ext. Valid characters are:

- uppercase and lowercase letters
- digits
- ! # \$ % & ' () + , - . ; = @ [] ^ _ ` { } ~ and space

The wildcard character * may be used for *device*. Both wildcard characters * and | may be used in *filename* and in .*ext*.

Syntax for x900-48FE and AT-9900 `SHoW FiLe[=filename] [DEvice={ALl|FLash|NVs|CFlash}]`

where *filename* is in the format [*device*:]*filename*.ext. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { } * > []

Wildcard characters * may appear in the filename when displaying them, but not when creating them. The wildcard character matches any string.

Character ranges may be specified using the > character, for example a>z matches any letter in the alphabet. The + character may be used to specify a list of options, for example a*.scp+b*.scp would specify files that match a*.scp or b*.scp.

Square brackets may be used, for example ppp*.[scp+cfg] matches scripts and configuration files whose names start with "ppp".

The vertical bar | character matches any single character. For example, | | |.scp matches script files with names three characters long (excluding extension and device name).

Description Depending on what is specified, this command displays one of the following:

- a list of all files in the file system ([Figure 6-12 on page 6-47](#), [Table 6-10 on page 6-47](#)) when no file name is specified
- a list of files in the file system that match a file name ([Figure 6-12](#), [Table 6-10](#)) when it is specified with wildcards
- the contents of a single file when a file name is specified without wildcards

This command requires a user with security officer privilege when the switch is in security mode.

The **device** parameter specifies the physical storage devices whose files are to be listed. This parameter is ignored if the file name includes a device name.

To display the contents of the translation table on x900-48FE and AT-9900 switches, use the **show file=longfile.lfn** command ([Figure 6-13 on page 6-48](#), [Table 6-11 on page 6-48](#)). The table converts filenames between DOS 28.3 format and DOS 8.3 format.

When a file is saved to CompactFlash, the directory structure allows duplicate filenames. For example, the files \subdir1\subdir2\filename.scp and \filename.scp have exact filenames but are different files. Uniqueness is provided by the directory structure. CompactFlash differentiates between the two files by the different directories they are in. The translation table does not save this directory information as it can get too large. A check value, based on the file directory string, is calculated, saved with the file data, and used in data retrieval.

Figure 6-11: Example output from the **show file** command for the x900-48FE and AT-9900 switches

Filename	Device	Size	Created	Locks
12345678901234567890.scp	flash	24	29-Mar-2004 15:34:21	0
13gggggg.scp	flash	8	29-Mar-2004 15:34:03	0
16a.scp	flash	7	17-Mar-2004 10:50:33	0
16abcd.scp	flash	14	17-Mar-2004 10:21:24	0
16ffff.scp	flash	32	16-Mar-2004 13:41:26	0
16ffffff.scp	flash	8	16-Mar-2004 14:17:19	0
409275.scp	flash	507	03-Nov-2003 12:07:37	0
409275a.scp	flash	441	24-Oct-2003 12:23:04	0
409451.scp	flash	588	10-Nov-2003 10:17:18	0
86263aka.rez	flash	3604528	16-Apr-2004 14:20:46	0
atobrsa.key	flash	321	04-Feb-2004 14:32:51	0
basic.cfg	flash	119	01-Dec-2003 15:35:56	0
bgp.cfg	flash	2811	15-Apr-2004 10:22:40	0
bgppeer.scp	flash	35	16-Apr-2004 09:59:20	0
cck.scp	flash	1018	14-Oct-2003 15:27:57	0
client.cfg	flash	2679	06-Nov-2003 13:38:48	0
config.ins	flash	32	19-Apr-2004 12:07:50	0

Figure 6-12: Example output from the **show file** command for the x900-24X switch

Filename	Device	Size	Created
AT9924s_301-00.pkg	flash	3351291	16-Mar-2005 12:20:48
base_pkg.lic	flash	300	09-Mar-2005 18:30:28
config.ins	flash	14	23-Feb-2005 17:15:08
config.scp	flash	40773	23-Feb-2005 17:04:52
prefer.ins	flash	1048	16-Mar-2005 12:21:08
showtimetest.scp	flash	141	23-Feb-2005 17:15:06
snmpEngn.sec	flash	40	18-Mar-2005 12:09:12
startup.cfg	nvs	3904	18-Mar-2005 12:29:22

Table 6-10: Parameters in output of the **show file** command

Parameter	Meaning
Filename	Name of the file.
Device	Device where the file is physically stored, such as flash.
Size	Size of the file in bytes as a decimal number.
Created	Date and time the file was created.
Locks	Number of concurrent processes using the file.

Figure 6-13: Example output from the **show file=longfile.lfn** command for x900-48FE and AT-9900 switches

short filename	device	long filename	created	size	check
123456~0.scp	flash	12345678901234567890.scp	15:34:21	24	0

Table 6-11: Parameters in output of the **show file=longfile.lfn** command

Parameter	Meaning
Short filename	Name of the file in DOS 8.3 format.
Device	Device where the file is physically stored, such as flash.
Long filename	Name of the file in DOS 28.3 format.
Created	Date and time the file was created.
Size	Size of the file in bytes, as a decimal number.
Check	For CFlash files the check value based on the file directory string. For flash and NVS files this value is 0 and not used.

Examples To display all patch files on an x90-48FE, use the command:

```
sh fi=*.paz
```

To display the contents of the config.scp script file, use the command:

```
sh fi=config.scp
```

To display a list of all package files on all storage devices on an x900-24X, use the command:

```
sh fi=*.pkg
```

To display a list of all files on a memory card on an x900-24X, use the command:

```
sh fi=card:
```

To display the contents of the longfile.lfn long filename table on an AT-9900, use the command:

```
sh fi=longfile.lfn
```

Related Commands

- [delete file](#)
- [show card](#)
- [purge file translationtable](#)
- [show flash](#)
- [show nvs](#)

show file permanentredirect

Syntax `SHoW FiLe[=filename] PERManentredirect`

Description This command displays information about one text file or all that are permanently receiving output from commands or scripts ([Figure 6-14](#), [Table 6-12](#)). These files are typically created to collect data during debugging.

The **file** parameter displays information about a specific text file ([Figure 6-15](#)). The *filename* option is in the format [device:]filename.txt and can be:

- uppercase and lowercase letters
- digits
- # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space

Device indicates the physical location where the file is stored. The default is flash.

Figure 6-14: Example output from the **show file permanentredirect** command

TTY Instance	Current Size	Limit	File
17	12345	204800	bgp.txt

Figure 6-15: Example output from the **show file=*filename* permanentredirect** command

File.....	bgp.txt
TTY Instance....	17
Current Size....	12345
Limit.....	204800
Input(s).....	COMMAND="enable bgp debug=all"

Table 6-12: Parameters in output of the **show file permanentredirect** command

Parameter	Meaning
TTY Instance	Instance number for the TTY device. For details about the TTY device, see the show tty command on page 46-14 of Chapter 46, Terminal Server .
Current Size	Size of the text file in bytes.
Limit	Limit of file size in bytes set by the limit parameter.
File	Name of text file.
Input(s)	Commands and scripts that generate input for the text file.

Examples To display all text files receiving output from commands or scripts, use the command:

```
sh fi perm
```

Related Commands [add file](#)
[create file](#)
[reset file permanentredirect](#)

show flash

Syntax for x900-24X SHow FLash

Syntax for x900-48FE and AT-9900 SHow FLash [FFs]

Description This command displays information about the file system stored in flash memory. The Flash File System (FFS) on x900-48FE and AT-9900 switches provides a consistent file-based interface to the physical flash memory structure, and housekeeping and management functions ([Figure 6-17](#), [Table 6-13](#)).

Figure 6-16: Example output from the **show flash** command on the x900-24X switch

```
Files total size .... 13832812 bytes (29 files)
Used space ..... 13875200 bytes
Free space ..... 14661632 bytes
Total space ..... 28536832 bytes
```

Figure 6-17: Example output from the **show flash** command on x900-48FE and AT-9900 switches

```
FFS info:
global operation ..... none
compaction count ..... 256
est compaction time ... 88 seconds
files ..... 1420044 bytes (4 files)
garbage ..... 19652 bytes
free ..... 526384 bytes
required free block ... 131072 bytes
total ..... 2097152 bytes

diagnostic counters:
event      successes      failures
-----
get         0             0
open        0             1
read        0             0
close       0             0
complete    0             0
write       0             0
create      0             0
put         0             0
delete      0             0
check       0             0
erase       0             0
compact     0             0
verify      0             0
-----
```

Table 6-13: Parameters in output of the **show flash** command

Parameter	Meaning
For x900-48FE and AT-9900 switches	
global operation	Global operation currently running; either none, restarting, erasing, compacting, or verifying.
compaction count	Number of times the flash has been compacted since the last total erasure.
est compaction time	Estimate of how long compaction would take if it was started now.
files	Amount of space used by valid files.
garbage	Amount of space used by deleted files.
free	Amount of free space.
required free block	Minimum contiguous working space. This amount of flash memory must remain available. Therefore, it is not included in the "free" entry.
total	Total flash size.
diagnostic counters	Counts of the successes and failures for each type of FFS operation.
For x900-24X switches	
Files total size	Total size of all files in flash with the number of files in parentheses.
Used space	Space that files use in flash. This figure is larger than the total size of the files because the file system has overhead, such as the last allocation cluster of each file being only partly filled.
Free space	Available space in flash.
Total space	Total capacity of the file system in flash.

FFS failure counts do not necessarily mean that an error has occurred, but are also incremented if the specified file could not be found. For example, attempting to delete a file that does not exist results in the delete failures count being incremented.

Related Commands [show card](#)
[show file](#)
[show nvs](#)

show flash physical

Syntax SHow FLash Physical

Description This command displays physical information about the specific type of flash installed in x900-48FE and AT-9900 switches (Figure 6-18, Table 6-14).

Figure 6-18: Example output from the **show flash physical** command

```
total size ..... 16 MBytes
  available to FFS ... 15 MBytes
  available to boot .. 1 MBytes
device type ..... 28F128
devices ..... 1
location ..... built in
programming power ..... off
block erase time ..... 1000 milliseconds
total erase blocks .... 128
  FFS erase blocks ... 120
  Boot erase blocks .. 8
erase block size ..... 128 kBytes
erase bit state ..... 1
page buffers ..... 1
size of page buffer ... 32 bytes
```

Table 6-14: Parameters in output of the **show flash physical** command

Parameter	Meaning
total size	Amount of flash memory installed.
available to FFS	Amount of flash memory available to the Flash Filing System.
available to boot	Amount of flash memory available to the boot flash.
device type	Type of flash device installed.
devices	Number of flash devices installed.
location	Whether flash memory is built in or a SIMM stick.
programming power	Whether programming power is on or off.
block erase time	Time taken to erase an erase block.
total erase blocks	Number of erase blocks.
FFS erase blocks	Number of erase blocks available to the Flash Filing System.
Boot erase blocks	Number of erase blocks available to the Boot system.
erase block size	Size of each erase block, in bytes.
erase bit state	State of an erased bit.
page buffers	Number of page buffers.
size of page buffer	Byte size of each page buffer.

Related Commands [show flash](#)

show nvs

Syntax for x900-24X `SHoW NVS`

Syntax for x900-48FE and AT-9900 `SHoW NVS [BLOCK=id [INDeX=index]]`

where:

- *id* is a block ID number in hexadecimal.
- *index* is a block index number in hexadecimal.

Description This command displays information about the file system stored in Non-Volatile Storage (NVS) memory on x900-48FE and AT-9900 switches (Figure 6-19, Table 6-15). It requires a user with security officer privilege when the switch is in security mode.

If the **block** parameter is specified, blocks with the specified ID are shown. If the **index** parameter is specified, the block with the specified ID and index are shown (Figure 6-20, Table 6-15).

Figure 6-19: Example output from the **show nvs** command on the x900-24X switch

```
Files total size ..... 3904 bytes (1 file)
Used space ..... 11264 bytes
Free space ..... 490496 bytes
Total space ..... 501760 bytes
```

Figure 6-20: Example output from the **show nvs** command on x900-48FE and AT-9900 switches

Block ID	Index	Size (bytes)	Creation Date	Creator ID	Block Address
0000001a	00000002	00000178	11-Aug-2000	00000012	ffe00200
0000001a	00000003	0000001a	01-Aug-2000	00000012	ffe00400
00000032	00000002	00000050	19-Feb-2001	00000022	ffe06000
00000038	00000000	00000000	01-Aug-2000	00000021	ffe00000
00000043	00000001	000000b4	27-Mar-2001	00000029	ffe02a00
00000043	00000002	00000f78	23-Mar-2001	00000029	ffe02e00
00000043	00000003	00000058	09-Mar-2001	00000029	ffe04c00
00000043	00000004	00000340	14-Nov-2000	00000029	ffe04200
00000043	00000005	00000340	14-Nov-2000	00000029	ffe04600
00000043	00000006	000004e8	20-Nov-2000	00000029	ffe06200
00000043	00000007	00000130	19-Dec-2000	00000029	ffe03000
00000043	00000008	00000088	21-Dec-2000	00000029	ffe08000
00000045	000003fc	00000de4	27-Mar-2001	00000026	ffe00a00
00000045	000003fd	00000024	27-Mar-2001	00000026	ffe00800

Table 6-15: Parameters in output of the **show nvs** command

Parameter	Meaning
For x900-48FE and AT-9900 switches	
Block ID	ID of the block in hexadecimal.
Index	Index of the block in hexadecimal.
Size (bytes)	Size of the block in hexadecimal bytes.
Creation Date	Date the block was created. "***_***_***" indicates that the date was undefined when the block was created.
Creator ID	ID of the module that created the block.
Block Address	Pointer to battery backed RAM where the block starts.
For x900-24X switches	
Files total size	Total size of all visible files in NVS with the number of visible files in parentheses.
Used space	Space that visible files use in NVS. This figure is larger than the total size of files because the file system has overhead, such as the last allocation cluster of each file being only partly filled, and also because the switch software creates invisible files for internal use.
Free space	Available space in NVS.
Total space	Total capacity of the file system in NVS.

Related Commands

- [delete nvs](#)
- [dump nvs](#)
- [modify nvs](#)
- [show card](#)
- [show file](#)
- [show flash](#)
- [show nvs free](#)

show nvs free

Syntax `SHoW NVS FRee`

Description This command shows how much free space there is in the non-volatile storage (NVS) on x900-48FE and AT-9900 switches. It also shows the size of the largest block that can be created ([Figure 6-21](#)).

Figure 6-21: Example output from the **show nvs free** command.

Number of free sectors	85
Number of bytes in free sectors	85656

Related Commands

- [clear nvs totally](#)
- [delete nvs](#)
- [dump nvs](#)
- [modify nvs](#)
- [show nvs](#)