

# Utility Commands


- utility cut, on page 2
- utility date, on page 6
- utility date set, on page 8
- utility fgrep, on page 9
- utility find, on page 11
- utility less, on page 13
- utility mv, on page 15
- utility sort, on page 16
- utility tail, on page 19
- utility uniq, on page 21
- utility wc, on page 23
- utility bc, on page 25
- utility df, on page 29
- utility du, on page 33
- utility egrep, on page 36
- utility head, on page 39
- utility which, on page 41
- utility xargs, on page 43

# utility cut

To extract selected characters or fields from standard input or from a file, use the **utility cut** command in XR EXEC mode.

**utility cut** { {**list** *character-list* | **fields** *field-list* [**nodelim**] [**delimiter** *delimiter-character*]*WORD*} [**file** *input-file*] | **usage**}

## Syntax Description

| | |
|---|---|
| **list** *character-list* | (**-c**) Cuts out the characters that are located on each line as specified with the *character-list* argument.<br><br>The *character-list* argument specifies the character positions or range of the characters to be cut.<br><br>• Use a comma (,) to indicate more than one character. For example, **utility list 1,2,5** outputs the first, second, and fifth characters.<br><br>• Use a dash (-) to indicate a range. For example, **utility list 1-64** outputs the first 64 characters of each line, **utility list 5-** outputs the fifth character to the end of the line.<br><br>**Note**<br>Lines are separated by a delimiter. The default delimiter is tab. |
| **fields** *field-list* | (**-f**) Cuts out the fields (lines) as indicated with the *field-list* argument.<br><br>The *field-list* argument specifies the field numbers or ranges. For example, **utility field 2,9** outputs the second and ninth fields, **utility field 1-3** outputs the first three fields, **utility field -6** outputs the first six fields.<br><br>**Note**<br>The fields indicated by the *field-list* argument are assumed to be separated in the file by a delimiter character. The default delimiter is tab. Use the **delimiter** *delimiter* option to specify a delimiter character. Lines without field delimiters are processed unless the **nodelim** keyword is specified. |
| **nodelim** | (Optional) (**-s**) Ignores lines with no delimiter. Use this optional keyword when the **fields** *field-list* keyword and argument is specified. |
| **delimiter** *delimiter-character* | (Optional) (**-d**) Specifies an alternative delimiter to indicate the end of each field. Replace the *delimiter-character* argument with the character used as the delimiter. |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **file** *input-file* | (Optional) Storage device and directory path of the text file used instead of the standard input (keyboard input).<br><br>The syntax of the *input-file* argument is: *device* **:**[*/ directory-path*]*/ filename*<br><br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |

| | |
|---|---|
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If no file is specified, **keyboard** input (standard input) is used.

The default for delimiter is **tab**.

## Command Modes

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The **utility cut** command cuts out columns, fields, or characters displayed from standard input or from a file.

Use the **fields** *field-list* keyword and argument if the fields vary in length from line to line. (The lines must be separated by a delimiter character.) By default, the field delimiter character is the Tab key. Use the **delimiter** *delimiter-character* keyword and argument to specify a different delimiter.

Use the **list** *character-list* keyword and argument only if the fields are of a fixed length. Replace the *character-list* argument with the character positions to be extracted.

For the *character-list* argument, use a comma (,) to indicate more than one character, or use a dash (-) to indicate a range. For example, **utility list 1,2,5** outputs the first, second, and fifth characters, **utility list 1-64** outputs the first 64 characters of each line, **utility list 5-** outputs the fifth character to the end of the line.

You can also use the cut utility as a filter. If no files are specified, the keyboard input (standard input) is used.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **fields** keyword can also be entered using the UNIX-equivalent (**-f**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility cut** command is entered with the **list** *character-list* keyword and argument to display the first 10 characters in each line. The output is from the results of the **show version** command, which is entered with the pipe (|) character:

```
RP/0/RP0/CPU0:router# show version | utility cut list 1-10

  Cisco IOS
  Copyright

  ROM: Syste

  router upt
  System ima

  cisco CRS-
  7457 proce

  16 Gigabit
  2 Ethernet
  20 Packet
  20 SONET/S
```

```
  2043k byte
  38079M byt
  1000592k b
  1000640k b

  Configurat
  Package ac
   --More--
```

In the following example, the **utility cut** command is used to extract fields from a file:

```
RP/0/RP0/CPU0:router# utility cut fields 1,5 delimiter : file disk0:/usr/passwd

  root:Super-User
  daemon:
  bin:
  sys:
  adm:Admin
  lp:Line Printer Admin
  uucp:uucp Admin
  nuucp:uucp Admin
  listen:Network Admin
  nobody:Nobody
```

In the following example, the **utility cut** command is used with the **delimiter** keyword to specify an alternative field delimiter:

```
RP/0/RP0/CPU0:router# utility cut fields 1,4,5 delimiter : file disk0:/usr/passwd

  root:1:Super-User
  daemon:1:
  bin:2:
  sys:3:
  adm:4:Admin
  lp:8:Line Printer Admin
  uucp:5:uucp Admin
  nuucp:9:uucp Admin
  listen:4:Network Admin
```

In the following example, a range of fields is specified:

```
RP/0/RP0/CPU0:router# utility cut fields 1-4 delimiter : file disk0:/usr/passwd

  root:x:0:1
  daemon:x:1:1
  bin:x:2:2
  sys:x:3:3
  adm:x:4:4
  lp:x:71:8
  uucp:x:5:5
  nuucp:x:9:9
  listen:x:37:4
```

In the following example, the **list** *character-list* keyword and argument are used to specify the character positions to be extracted:

```
RP/0/RP0/CPU0:router# utility cut list 1-30 file disk0:/usr/passwd
```

```
  root:x:0:1:Super-User:/:/sbin/
  daemon:x:1:1::/:
  bin:x:2:2::/usr/bin:
  sys:x:3:3::/:
  adm:x:4:4:Admin:/var/adm:
  lp:x:71:8:Line Printer Admin:/
  uucp:x:5:5:uucp Admin:/usr/lib
  nuucp:x:9:9:uucp Admin:/var/sp
  listen:x:37:4:Network Admin:/u
  nobody:x:60001:60001:Nobody:/:
  noaccess:x:60002:60002:No Acce
  nobody4:x:65534:65534:SunOS 4.

  ================================================================================
```

In the following example, the UNIX equivalent options are used directly. First, the **utility cut** command is entered with the **usage** keyword to display the possible options. Next, the **utility cut** command is entered with the options to extract the desired data.

```
RP/0/RP0/CPU0:router# utility cut usage

  cut -c list [file], cut -f list [-d delim] [-s] [file]

RP/0/RP0/CPU0:router# utility cut -f 1,4 -d : disk0:/usr/passwd

  root:1
  daemon:1
  bin:2
  sys:3
  adm:4
  lp:8
```

# utility date

To display the date and time, use the **utility date** command in XR EXEC mode.

**utility date** {**format** *word* | **universal** | **usage***WORD*}

## Syntax Description

| | |
|---|---|
| **format** *word* | (Optional) (+) Specifies the format for the date display. Use the online help system to display the available format syntax for the *word* argument. |
| **universal** | (Optional) (**-u**) Displays the date in Coordinated Universal Time (UTC) instead of local time. UTC is the standard term for Greenwich Mean Time (GMT). |
| **usage** | (Optional) Displays the UNIX options supported by this command. |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |

## Command Default

The date is displayed in local time.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The **utility date** command displays the internal time and date for the router.

### Date Format

Use the **format** *word* option to specify the format and content of the displayed date and time. The format is composed of ASCII characters and field descriptors prefaced with %, in a manner similar to a C-language printf() format specifier. In the output, each field descriptor is replaced by its corresponding value; all other characters are copied to the output without change. The format is specified using the following characters:

**%C**

Century in 'CC' form. For example: 20

**%y**

Year in 'YY' form. For example: 06

**%m**

Month in 'MM' form. For example: 08

**%d**

Date in 'DD' form. For example: 28

**%H**

Hour in 'hh (24 hr.)' form. For example: 18

**%M**

Minutes in 'mm' form. For example: 55

**%S**

seconds in 'ss' form. For example: 24

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **universal** keyword can also be entered using the UNIX-equivalent (**-u**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

This example shows how to display the router date and time using the **utility date** command:

```
RP/0/RP0/CPU0:router# utility date

  Fri Aug 04 11:53:38 UTC 2006
```

This example shows how to display the router date and time using a variety of options with the **format** keyword:

```
RP/0/RP0/CPU0:router# utility date format "%y%m%d"

  060828

RP/0/RP0/CPU0:router# utility date format "%y-%m-%d"

  06-08-28

RP/0/RP0/CPU0:router# utility date format "%C%y-%m-%d"

  2006-08-28

RP/0/RP0/CPU0:router# utility date format "%C%y-%m-%d:%H:%M:%S"

  2006-08-28:02:09:58

RP/0/RP0/CPU0:router# utility date format "DATE: %y-%m-%d %nTIME: %H:%M:%S"

  DATE: 06-09-17
  TIME: 12:42:24
```

# utility date set

To set the router time, use the **utility date set** command in System Admin EXEC mode.

**utility date set** *hh*:*mm*:*ss*

**Syntax Description**

| | |
|---|---|
| *hh* | Specifies the hour in 2-digit numerical format. Range is 00 to23. |
| *mm* | Specifies the minutes in 2-digit numerical format. Range is 0 to 59. |
| *SS* | Specifies the seconds in 2-digit numerical format. Range is 0 to 59. |

**Command Default**

None

**Command Modes**

System Admin EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

**Usage Guidelines**

A colon (:) is required between the entry for hour, minutes, and seconds.

**Note** Generally, if the system is synchronized by a valid outside timing mechanism, such as a Network Time Protocol (NTP) clock source, or if you have a networking device with calendar capability, you need not set the software clock. Use the **date** command or the **clock set** command if no other time sources are available.

**Note** To manually copy the hardware clock (calendar) settings into the software clock, use the **clock read-calendar** command in EXEC mode.

By default, the system makes a "slow adjustment" if the new time is in the range of the following:

- –2.5 minutes + old time
- 5 minutes + old time

In a slow adjustment, the clock speed increases by less than 100 percent or decreases by less than 50 percent over a period of time from 1 second to 5 minutes until the clock catches up with the new time. This slow adjustment does not cause major discontinuities in the time flow. Use the **-S0** option to disable the slow adjustment.

The following example shows how to set the time using the **utility date set** command:

```
RP/0/RP0/CPU0:router(admin)# utility date set 13:07:00

  Fri Sep 15 13:07:00 UTC 2006
```

# utility fgrep

To search a file for a fixed character string, use the **utility fgrep** command in XR EXEC mode.

**utility fgrep** {**expr** *expression* | **script** *expression-file*} [*WORD*] [**count**] [**linenum**] [**matchfile**] [**matchline**] [**nocase**] [**nofile**] [**reverse**] [**file** *search-file*]

**utility fgrep** *expression* [*WORD*] [**count**] [**linenum**] [**matchfile**] [**matchline**] [**nocase**] [**nofile**] [**reverse**] [**file** *search-file*]

**utility fgrep usage**

## Syntax Description

| | |
|---|---|
| **expr** *expression* | (**-e**) A regular expression, whose type is determined by the -e and -f options. This form is used when only one expression is specified on the command line. Any names specified after this option are treated as input files. |
| **script** *expression-file* | (**-f**) A file containing a set of regular expressions, each separated by a new line. The type of the expressions is determined by the -e and -f options. This form is used when more than one expression is specified. You can specify more than one -f option.<br>The syntax of the *expression-file* argument is: *device* **:**[*/ directory-path*]*/ filename* |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 20. |
| **count** | (Optional) (**-c**) Displays a count of selected lines. |
| **linenum** | (Optional) (**-n**) Before each output line, displays the line's line number. |
| **matchfile** | (Optional) (**-l**) (“el”) Displays only the names of files containing the selected lines. |
| **matchline** | (Optional) (**-x**) Includes only input lines selected against an entire fixed string or regular expression. |
| **nocase** | (Optional) (**-i**) Ignores uppercase and lowercase distinctions during comparisons. |
| **nofile** | (Optional) (**-h**) Displays results without a filename prefix attached to the matched lines. This option applies only when more than one file is searched. |
| **reverse** | (Optional) (**-v**) Selects only those lines that don't match the specified patterns. |
| **file** *search-file* | (Optional) The file used for the search. Replace the *search-file* argument with the device and directory path of the file. The syntax for the *search-file* argument is: *device* **:**[*/ directory-path*]*/ filename* |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

The keyboard input (standard input) is used if no files are specified.

If more than one input file is specified, then the filename is displayed before each line.

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The **utility fgrep** command searches files for a fixed character string (as opposed to grep and egrep, which search for a a pattern that matches an expression).

The results are displayed to the standard output (terminal screen).

**Note** The fgrep utility options are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **count** keyword can also be entered using the UNIX-equivalent (**-c**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

The following example, the **utility fgrep** command is used with the **nocase** and **linenum** keywords:

```
RP/0/RP0/CPU0:router# show version | utility fgrep expr uptime nocase linenum

  7:router uptime is 5 days, 20 hours, 10 minutes
```

# utility find

To locate files within one or more directories, use the **utility find** command in XR EXEC mode.

**utility find** {**path** *directory-path* {*LINE* | **name** *filename-pattern* | **user** *user-id*} | **usage**}

## Syntax Description

| | |
|---|---|
| **path** *directory-path* | Specifies the storage device and directory for the file search. The search is performed for the specified directory and all subdirectories in that directory tree.<br>If a directory path is not specified, then the search is performed in the current directory (a path of . [dot] is assumed). |
| *LINE* | (Optional) UNIX command-line expressions provided as a string. |
| **name** *filename-pattern* | (Optional) Searches for the name of the file. The *filename-pattern* argument is a regular expression string. |
| **user** *user-id* | (Optional) Searches for files belonging to a specific user. The *user-id* argument is the username of the file owner. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If a directory path is not specified, then the search is performed in the current directory.

If a **name** *filename-pattern* is not specified, then the search return all files in the specified directory.

If a user is not specified, then the search is performed for all users.

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Use the **utility find** command to locate files within one or more directories. You can perform the search for a specific directory (and its subdirectories). If a directory is not specified, then the search is performed for the current directory.

To search for a regular expression string, use the **name** *filename-pattern* keyword and argument. Replace the *filename-pattern* argument with the regular expression string. If this option is not used, then all files within the specified directory are displayed.

To search for files belonging to a specific user, use the *user-id* argument. If this option is not used, then files belonging to all users are displayed.

In the following example, the **utility find** command is used to locate files matching a pattern. In this example, all files ending in ".txt" are displayed:

```
RP/0/RP0/CPU0:router# utility find path disk0:/usr name *.txt
```

```
disk0:/usr/test2.txt
```

In the following example, the UNIX equivalent option is used to locate files matching a pattern. In this example, all files ending in ".txt" are displayed:

```
RP/0/RP0/CPU0:router# utility find path disk0: -name *.txt

disk0:/-base-3.8.0/etc/vim/doc/editing.txt
disk0:/-base-3.8.0/etc/vim/doc/help.txt
disk0:/-base-3.8.0/etc/vim/doc/intro.txt
disk0:/-base-3.8.0/etc/vim/doc/uganda.txt
disk0:/usr/test2.txt
```

In the following example, the files belonging to a specific user are displayed:

```
RP/0/RP0/CPU0:router# utility find path disk0:/usr user 0

disk0:/usr
disk0:/usr/passwd
disk0:/usr/test2.txt
```

In the following example, the UNIX equivalent option is used to display files belonging to a specific user:

```
RP/0/RP0/CPU0:router# utility find path disk0:/usr -user 0

disk0:/usr
disk0:/usr/passwd
disk0:/usr/test2.txt
```

# utility less

To display a file page-by-page, use the **utility less** command in XR EXEC mode.

**utility less** {[**exitEOF**] [*WORD*] | **nocase** | **position** *line-number* | **startat** *string*} [**file** *source-file*]

**Syntax Description**

| | |
|---|---|
| **exitEOF** | (Optional) (**-E**) Automatically exits the utility the first time an end-of-file is encountered. |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **nocase** | (Optional) (**-i**) Ignores uppercase and lowercase distinctions during comparisons. |
| **position** *line-number* | (Optional) (**-j**) Uses the line at *line-number* on the screen to position matched lines during a patter search. |
| **startat** *string* | (Optional) (**-p**) Starts at the first occurrence of the pattern specified by the *string* argument in the file. |
| **file** *source-file* | (Optional) Specifies the storage device and directory path for the text file to be displayed. The default is standard input.<br>The syntax for the *source-file* argument is: *device* **:**[/ *directory-path*]/ *filename* |

**Command Default**

If no text file is specified, standard input is assumed.

**Command Modes**

XR EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

**Usage Guidelines**

Use the **utility less**command to display files page by page. You can specify regular expressions for pattern matching using the **startat** keyword. You can scroll up as well as down. When you enter the less mode, commands are similar to the “vi” editor.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

The following example, the **utility less** command is used to display the file “config_store”. Only part of the file is shown here.

```
RP/0/RP0/CPU0:router# utility less file disk0:/usr/config_store

  Last configuration change at Tue Feb 20 18:34:02 2007 by xxx
```

```
!
hostname H1
line console
 exec-timeout 600 0
 session-timeout 600
!
line default
 exec-timeout 600 0
 session-timeout 600
!
.
.
.
```

# utility mv

To rename or move a file from one directory to another, use the **utility mv** command in XR EXEC mode.

**utility mv** {[*WORD* | **force** | **interactive**] **source** *source-file* **target** *target-file* | **usage**}

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **force** | (Optional) (**-f**) Forces an overwrite if the target file already exists. There is no confirmation prompt. |
| **interactive** | (Optional) (**-i**) Specifies to prompt for confirmation before renaming a file. |
| **source** *source-file* | Specifies the storage device, directory, and filename for the file to be moved. |
| **target** *target-file* | Specifies the new storage device, directory, and filename for the file. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

No default behavior or values

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility mv** command is used to move the file “aaa” from disk0a: to disk1a:

```
RP/0/RP0/CPU0:router# utility mv source disk0a:/aaa target disk1a:/aaa
```

# utility sort

To sort, merge, or sequence-check the lines in one or more files, or from the standard input, use the **utility sort** command in XR EXEC mode.

**utility sort** {[[*WORD*] | [[**dict**] [**fieldSep** *character*] [**ignoreblank**] [**key** *key-definition*] [**lowercase**] [**merge**] [**numeric**] [**outfile** *filename*] [**printable**] [**reverse**] [**unique**]]] [**file** *filename*] | **usage**}

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **dict** | (Optional) (**-d**) Sorts in dictionary order. Uses only alphanumeric and blank characters in the sort operation. |
| **fieldSep** *character* | (Optional) (**-t**) Specifies a character as the field separator. |
| **ignoreblank** | (Optional) (**-b**) Ignores leading blank characters in field comparisons. |
| **key** *key-definition* | (Optional) (**-k**) Defines a key to be the sort key. The *key-definition* argument field is defined using the following syntax:<br>*field_start* [*type_string*] [,*field_end*] [*type_string*]<br>• *field_start*<br>and *field_end*—Specifies the beginning and end of the key field.<br>• *type_string*—Specifies attributes specific to the key.<br>The *field_start* and *field_end* arguments are each specified by a pair of digits of the form m.n, where the m refers to the field starting after the mth field separator in a line. For field_start, the .n refers to the nth character of the specified field, and is taken as zero if not specified. For field_end, the .n refers to the nth character after the last character of the specified field, and is taken as zero if not specified.<br>The *type_string* argument may be formed from the characters bdfinr, which apply their defined attributes to the determination of the key.<br>**Note**<br>When ordering options appear independent of key field specifications, the requested field ordering rules are applied globally to all sort keys. When attached to a specific key, the specified ordering options override all global ordering options for that key. |
| **lowercase** | (Optional) (**-f**) Folds uppercase letters into lowercase (ignores case and treats upper case characters the same as lowercase characters). |
| **merge** | (Optional) (**-m**) Merges sorted files. Assumes that the files are already sorted and so does not sort the files. |
| **numeric** | (Optional) (**-n**) Interprets the field as numeric and sorts in numeric order. Includes the sign and optional thousands separator. This keyword also ignores leading blank characters in field comparisons (implies the **ignoreblank** keyword). |

| | |
|---|---|
| **outfile** *filename* | (Optional) (**-o**) Writes the results to a file. The *filename* argument is the destination disk, directory, and filename. The *filename* argument can be the same as the source file. |
| **printable** | (Optional) (**-i**) Ignores all nonprintable characters. |
| **reverse** | (Optional) (**-r**) Reverses the sort order. The sort is ascending by default. |
| **unique** | (Optional) (**-u**) Suppresses all but one line in each set of lines having equal keys. |
| **file** *filename* | (Optional) Specifies a file to be sorted. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If no file is specified, then the standard input (keyboard) is used.

If an **outfile** *filename* keyword and argument is not specified, then the standard output (display) is used.

The file is sorted in ascending order.

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility sort** command is used to sort the contents of the file “words.txt”:

```
RP/0/RP0/CPU0:router# utility sort file disk0:/usr/words.txt

   The
   few
   inquires
  A
  Code.
  Date
  Done
  This
  best-selling
  bestseller
  book
  come
  concerning
  fiction,
  have
  its
```

```
list
muscled
of
onto
our
the
way
way
work
```

# utility tail

To copy the end portion of a file or the standard input, use the **utility tail** command in XR EXEC mode.

**utility tail** {[[*WORD*] | [**bytes**] [**continuous**] [**count** *number*]] [**file** *input-file*] | **usage**}

**Syntax Description**

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **bytes** | (Optional) (**-c**) Copies the end of the file measured in bytes. The default is lines. |
| **continuous** | (Optional) (**-f**) Continues to copy data from the end of the file after the last line is reached. The operation pauses for 1 second, and then resumes in a continuous loop.<br>The input file must be a regular file, not a terminal or a FIFO special file (a named pipe). |
| **count** *number* | (Optional) (**-n**) Copies the number of lines (default) or bytes specified with the *number* argument. The range is 0 to 4294967295. By default, the last 10 lines are copied.<br>The *number* argument is a decimal integer that defines the location in the file to begin copying:<br>• Include the plus (+) character to copy from the beginning of the file.<br>• Include the minus (-) character to copy from the end of the file.<br>• Do not include a character to copy from the end of the file.<br>**Note**<br>Select the **bytes** keyword to copy the information measured in a count of bytes. |
| **file** *input-file* | (Optional) Directory path and filename for the input file. If no file is specified, then the standard input is used.<br>The syntax for the *input-file* argument is: *device* **:**[*/ directory-path*]*/ filename*<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

**Command Default**

If the **utility tail** command is entered without keywords or arguments, the last 10 lines of the standard input are copied.

**Command Modes**

XR EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

**Usage Guidelines**

Use the **utility tail** command to copy data from the end of a file. By default, the last 10 lines are copied. Use the **bytes** keyword to copy the data measured in bytes. Use the **count** *number* option to define the number of lines or bytes to copy. Use the **file** *filename* option to specify an input file.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility tail** command is used with the bytes keyword to display the last 10 bytes in the output:

```
RP/0/RP0/CPU0:router# show version | utility tail count 10 bytes

  .95.3-p8

RP/0/RP0/CPU0:router#
```

# utility uniq

To display or remove repeated lines in a file, use the **utility uniq** command in XR EXEC mode.

**utility uniq** [[[*WORD*] | [**afterChars** *number*] [**afterField** *number*] [**count**] [**nonrepeating** | **repeating**]] [**infile** *input-file* **outfile** *output-file*] | **usage**]

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **afterChars** *number* | (Optional) (**-s**) Ignores the first characters on each line of the input file. Use the *number* argument to specify the number of characters. The range is 0 to 4294967295. |
| **afterField** *number* | (Optional) (**-f**) Ignores the first fields on each line of the input file. Use the *number* argument to specify the number of fields. The range is 0 to 4294967295. |
| **count** | (Optional) (**-c**) Displays the number of times the line appeared in the input file at the beginning of each output line. |
| **nonrepeating** | (Optional) (**-u**) Displays only the nonrepeating lines from the input file (repeating lines are not displayed). |
| **repeating** | (Optional) (**-d**) Displays only the repeating lines from the input file (nonrepeating lines are not displayed). |
| **infile** *input-file* | (Optional) Specifies an input file for processing. The *input-file* argument specifies the device, directory, and filename of the input file. If no input file is specified, then the standard input (keyboard) is used.<br>The syntax of the *input-file* argument is: *device* **:**[*/ directory-path*]*/ filename*.<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |
| **outfile** *output-file* | (Optional) Specifies an output file. The *output-file* argument specifies the device, directory, and filename of the output file. If no file is specified, then the standard output (display) is used.<br>The syntax of the *output-file* argument is: *device* **:**[*/ directory-path*]*/ filename*.<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If no input file is specified, then the standard input is used.

If no output file is specified, then the standard output is used.

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Use the **utility uniq** command to display only lines that are repeated in a file, or to display only lines that appear once. This utility compares only adjacent lines, so the file or standard input must be sorted.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility uniq** command is used to display the repeating lines in the output of the **show environment** command:

```
RP/0/RP0/CPU0:router# show environment | utility uniq repeating

host    5V              4500,5500       4250,5750       4000,6000
fabricq 1.25V           1125,1375       1063,1438       1000,1500
fabricq 1.25V           1125,1375       1063,1438       1000,1500
ingress 1.25V           1125,1375       1063,1438       1000,1500
spa5    1.5V            1500,0          1575,1425       0,0
host    5V              4500,5500       4250,5750       4000,6000
fabricq 1.25V           1125,1375       1063,1438       1000,1500
fabricq 1.25V           1125,1375       1063,1438       1000,1500
ingress 1.25V           1125,1375       1063,1438       1000,1500
spa5    1.5V            1500,0          1575,1425       0,0
```

# utility wc

To count words, lines, or bytes in a file, use the **utility wc** command in XR EXEC mode.

**utility wc** [[[*WORD*] | [**bytes**] [**lines**] [**words**]] [**file** *input-file*] | **usage**]

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **bytes** | (Optional) (**-c**) Displays the number of bytes in each input file. |
| **lines** | (Optional) (**-l**) (-œel-?) Displays the number of lines in each input file. |
| **words** | (Optional) (**-w**) Displays the number of words in each input file. |
| **file** *input-file* | (Optional) Specifies the input file. The *input-file* argument specifies the device, directory, and filename of the input file. If no input file is specified, then the standard input (keyboard) is used.<br>The syntax of the *input-file* argument is: *device* **:**[*/ directory-path*]*/ filename*.<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

Output is displayed in the order bytes, words, and lines, even if the options are entered in a different order.

## Command Modes

XR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Output is displayed in the following order:

- When keywords are entered, the output appears in the order bytes, words, and lines.
- When no keyword is entered, the output appears in the order lines, words, and bytes.
- When any UNIX equivalent options are entered, the output appears in the order specified by the options. For example, if the command **utility wc -w -l -c** is entered, the output appears in the order words, lines, and bytes.



**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility wc**command is issued to display the number of lines, words, and bytes in the output of the **show version** command:

```
RP/0/RP0/CPU0:router# show version | utility wc

        221     1160    10820
```

The output displays the following:

- 221 lines
- 1160 words
- 10820 bytes

In the following example, the **utility wc** command is entered with the **words** keyword to display the number of words in the output of the **show version** command:

```
RP/0/RP0/CPU0:router# show version | utility wc words

       1160
```

# utility bc

To implement an arbitrary precision calculator, use the **utility bc** command in XR EXEC mode.

**utility bc** [**file** *input-file*]

## Syntax Description

**file** *input-file* (Optional) Specifies the text file containing commands and function definitions to be interpreted by the bc utility.

After all files have been read, the bc utility reads input from the standard input (keyboard). If no files are specified, then only the standard input (keyboard) is used.

The syntax of the *input-file* argument is as follows: *device* **:**[*/ directory-path*]*/ filename*

Possible values of the *device:* argument are:

**disk0:**

Uses a file from disk0: file system.

**disk0a:**

Uses a file from disk0a: file system partition.

**disk1:**

Uses a file from disk1: file system.

**disk1a:**

Uses a file from disk1a: file system partition.

**ftp:**

Uses a file from an FTP network server. The syntax is **ftp:**[[[//*username*[:*password*]@]*location*]/*directory*]/*filename*

**harddisk:**

Uses a file from the hard disk drive file system (if present).

**harddiska:**

Uses a file from the hard disk partition (if present).

**nvram:**

Uses a file from the nvram: file system.

**ipv4**

Uses a file from an IPv4 access list or prefix list.

**ipv6**

Uses a file from an IPv6 access list or prefix list.

**rcp:**

Uses a file from a remote copy protocol (rcp) network server. The syntax is **rcp:**[[[//*username*@]*location*]/*directory*]/*filename*

**tftp:**

Uses a file from a TFTP network server. The syntax is **tftp:**[[//*location*]/*directory*]/*filename*

Use the online help (**?**) function to display the available devices and network protocols.

## Command Default

If an input file is not specified, the standard input (keyboard) is used.

**Command Modes**

XR EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

**Usage Guidelines**

Use the **utility bc** command to use the interactive, programmable calculator that supports a complete set of control structures, including functions. The utility first processes any specified files, and then reads input from the keyboard (standard input).

Input files (and standard input) are ASCII text files containing sequences of bc statements to be executed.



**Note** The bc utility uses the bc programming language, an arbitrary precision calculator language with syntax similar to the C programming language. The bc utility does not support character or string manipulation.

The bc utility supports:

- 26 functions
- 26 simple variables
- 26 array variables (up to 2048 elements per array).

The bc utility supports the following common programming language constructs:

- “if”, “while”, and “for” statements
- User-defined functions with parameters
- Local variables

### Information About Supported Network Protocols

In the syntax for the **ftp:**, **rcp:**, and **tftp:** network protocols, the location is either an IP address or a hostname. The filename is specified relative to the directory used for file transfers.

When no device is specified, the current directory is used. To view the current directory, enter the **pwd** command.

***Table 1: Network Protocols Supported by Cisco IOS XR Software***

| Prefix | Name | Description |
|---|---|---|
| **tftp:** | Trivial File Transfer Protocol | *TFTP* is a simplified version of FTP that allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password). |
| **ftp:** | File Transfer Protocol | *FTP* is an application protocol, part of the TCP/IP protocol stack, and is used for transferring files between network nodes. FTP requires a username and password. |

| Prefix | Name | Description |
|---|---|---|
| **rcp:** | remote copy protocol | *Rcp* is a protocol that allows users to copy files to and from a file system residing on a remote host or server on the network. Rcp uses TCP to ensure the reliable delivery of data. Rcp downloads require a username. |

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the **utility bc** command is used to execute the bc statements contained in the ASCII text file exp.txt:

```
RP/0/RP0/CPU0:router# utility bc file disk0:/usr/exp.txt

50
15
25
3
17
```

# utility df

To display the amount of disk space available for a directory or file, use the **utility df** command in XR EXEC mode.

**utility df** [[*WORD*] [**kbytes**] [**mountinfo**] [**vsfStats**] [**file** *input-file*] | **usage**]

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **kbytes** | (Optional) (**-k**) Displays the sizes in 1-K blocks (1024-byte units) instead of the default of 512 byte blocks. |
| **mountinfo** | (Optional) (**-n**) Displays the file-system mountpoints and types only. |
| **vsfStats** | (Optional) (**-g**) Displays all statvfs() information. |
| **file** *input-file* | (Optional) Specifies the storage device and directory path of the device, directory, or file. When a directory or file is specified, the df utility displays the amount of space on the file system that contains the directory or file.<br>If no files are specified, then only the standard input (keyboard) is used.<br>The syntax of the *input-file* argument is as follows: *device* **:**[*/ directory-path*]*/ filename*<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

Information is displayed for all file systems.

The results are displayed in 512-byte blocks.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Use the (disk free) **utility df** command to display the amount of disk space available for a device, directory, or file. Enter the command without keywords or arguments to display information for all mounted file systems.

Use the **vsfStats** keyword to invoke the statvfs() function, which provides additional details for all mounted file systems.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **kbytes** keyword can also be entered using the UNIX-equivalent (**-k**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the (disk free) **utility df** command is entered without keywords or arguments to display information for all file systems:

```
RP/0/RP0/CPU0:router(admin)# utility df

  /dev/hd0t6              77987744     61592  77926152       1%  /harddisk:/
  /nvram:                     4086        60      4026       2%
  /dev/disk1t6             2001280    382720   1618560      20%  /disk1:/
  /dev/disk0t6             2001184    533568   1467616      27%  /disk0:/
  /dev/fs0p1                121856        68    121787       1%  /bootflash:
```

See Table 3: utility df Column Descriptions (left to right), on page 31 for column descriptions.

In the following example, the **kbytes** keyword is used to display information in 1-K blocks (1024-byte units) instead of the default of 512-byte blocks:

```
RP/0/RP0/CPU0:router(admin)# utility df kbytes

  /dev/hd0t6              38993872     30796  38963076       1%  /harddisk:/
  /nvram:                     2043        30      2013       2%
  /dev/disk1t6             1000640    191360    809280      20%  /disk1:/
  /dev/disk0t6             1000592    266784    733808      27%  /disk0:/
  /dev/fs0p1                 60928        34     60893       1%  /bootflash:
```

See Table 3: utility df Column Descriptions (left to right), on page 31 for column descriptions.

In the following example, the **mountinfo** keyword is used to display file-system mountpoints and types only:

```
RP/0/RP0/CPU0:router(admin)# utility df mountinfo

  Filesystem           Mounted on        Type
  /dev/hd0t6           /harddisk:/       dos (fat32)
  /nvram:
  /dev/disk1t6         /disk1:/          dos (fat16)
  /dev/disk0t6         /disk0:/          dos (fat16)
  /dev/fs0p1           /bootflash:       flash
```

In the following example, the **vfsStats** keyword is used to invoke the statvfs() function, which provides additional details for all mounted file systems:

```
RP/0/RP0/CPU0:router(admin)# utility df vfsStats

  /dev/hd0t6   /harddisk:/
      Blocks:  9748468 total   9740769 avail   [4096-byte blocks]
      Files :        0 total         0 avail
      Type  : dos (fat32)
      Flags : 00000120 [32bit, noatime]

  /nvram:
```

```
    Blocks:     2043 total      2013 avail   [1024-byte blocks]
    Files :        0 total         0 avail
    Type  :
    Flags : 00000000 []

/dev/disk1t6   /disk1:/
    Blocks:    62540 total     50580 avail   [16384-byte blocks]
    Files :        0 total         0 avail
    Type  : dos (fat16)
    Flags : 00000120 [32bit, noatime]

/dev/disk0t6   /disk0:/
    Blocks:    62537 total     45863 avail   [16384-byte blocks]
    Files :        0 total         0 avail
    Type  : dos (fat16)
    Flags : 00000120 [32bit, noatime]

/dev/fs0p1   /bootflash:
    Blocks: 62390272 total  62355240 avail   [1-byte blocks]
    Files :        2 total         0 avail
    Type  : flash
    Flags : 00000000 []
```

This table describes the significant fields shown in the display.

***Table 2: utility df Field Descriptions***

| Field | Description |
|---|---|
| Files | Number of files in the file system. |
| Blocks | Amount of space available on the file system in 1-K blocks. |
| Total | Amount of disk space used by the directory or file. |
| Avail | Amount of space available for use by the directory or file on the file system. |
| Type | Type of file system. |
| Flags | Displays the file system properties. |

In the following example, the **file** *source* keyword and argument are used to specify a directory:

```
RP/0/RP0/CPU0:router(admin)# utility df file disk0:/usr

  /dev/disk0t6             2001184    533568   1467616      27%  /disk0:/
```

This table describes the significant fields shown in the display.

***Table 3: utility df Column Descriptions (left to right)***

| Field | Description |
|---|---|
| Filesystem | File system for the displayed information. |
| 1k-blocks | Amount of space available on the file system in 1-K blocks. |

| Field | Description |
|---|---|
| Used | Amount of disk space used by the directory or file. |
| Available | Amount of space available for use by the directory or file on the file system. |
| Use% | Percentage of space used on the file system. |
| Mounted on | Storage device where the file system is mounted. |

## Related Commands

| Command | Description |
|---|---|
| utility du, on page 33 | Displays the amount of disk space used by one or more directories or files. |

# utility du

To display the amount of disk space used in a device, directory, or file, use the **utility du** command in XR EXEC mode.

**utility du** [[**all** | **specified**] [**kbytes** | **bytes** | **local**] [*WORD*] [**file** *source*] | **usage**]

## Syntax Description

| | |
|---|---|
| **all** | (Optional) (**-a**) Displays the disk space used for each file in the directory. By default, information is displayed only for the directory. Use the **all** keyword to display the total disk space used by all files in the directory, including the directory itself. |
| **specified** | (Optional) (**-s)** Displays the total disk space used for each specified file, rather than the totals for any subdirectories. |
| **kbytes** | (Optional) (**-k**) Displays the disk space used in 1-K blocks (1024-byte units) instead of the default of 512-byte blocks. |
| **bytes** | (Optional) (**-p**) Displays the disk space used in bytes (the default is 512-byte blocks). Also generates error messages for exiting files that cannot be displayed. |
| **local** | (Optional) (**-x**) Displays information for the local device only. |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **file** *source* | (Optional) Displays the disk space used for a device, directory, or file.<br>The syntax for the *source* argument is *device* **:** / *directory-path*[/ *filename*]<br>The *device* argument, followed by a colon, indicates the name of the device where the file is located. Use the online help (**?**) function to display the available storage devices and network protocols.<br>Enter the **utility du** command without specifying a device, directory, or file to display information for the current directory, and subdirectories. (The command behaves as if the filename dot (.) is entered.) |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

Information for the current directory is displayed.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Enter the **utility du** command without specifying a file to display information for the current directory. The command behaves as if the filename dot (.) is entered.

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **kbytes** keyword can also be entered using the UNIX-equivalent (**-k**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the **utility du** command is used to display the disk space used for the subdirectories in the MPLS package directory:

```
RP/0/RP0/CPU0:router# utility du file disk1:/asr9k-mpls-3.9.0.14I

Tue Jul 28 03:20:34.059 DST
         1607 /disk1:/asr9k-mpls-3.9.0.14I/schema
           83 /disk1:/asr9k-mpls-3.9.0.14I/lib/cerrno
          944 /disk1:/asr9k-mpls-3.9.0.14I/lib/mib
         3434 /disk1:/asr9k-mpls-3.9.0.14I/lib
        15974 /disk1:/asr9k-mpls-3.9.0.14I/bin
           14 /disk1:/asr9k-mpls-3.9.0.14I/mib
           14 /disk1:/asr9k-mpls-3.9.0.14I/placement
           14 /disk1:/asr9k-mpls-3.9.0.14I/startup
         1099 /disk1:/asr9k-mpls-3.9.0.14I/parser
           18 /disk1:/asr9k-mpls-3.9.0.14I/configs
           11 /disk1:/asr9k-mpls-3.9.0.14I/rules
           14 /disk1:/asr9k-mpls-3.9.0.14I/partitions
          156 /disk1:/asr9k-mpls-3.9.0.14I/etc/compat
          164 /disk1:/asr9k-mpls-3.9.0.14I/etc
           59 /disk1:/asr9k-mpls-3.9.0.14I/instdb_v
          147 /disk1:/asr9k-mpls-3.9.0.14I/lc/bin
            9 /disk1:/asr9k-mpls-3.9.0.14I/lc/startup
          164 /disk1:/asr9k-mpls-3.9.0.14I/lc
          103 /disk1:/asr9k-mpls-3.9.0.14I/instdb
        22697 /disk1:/asr9k-mpls-3.9.0.14I
```

In the following example, the **utility du** command is used to display the disk space used for a subdirectories:

```
RP/0/RP0/CPU0:router# utility du file disk0:/

-mpls-3.8.0/configs37 /disk0:/-mpls-3.8.0/configs
```

In the following example, the **utility du** command is used to display the disk space used for the current working directory:

```
RP/0/RP0/CPU0:router# cd disk0:/

-mpls-3.8.0
RP/0/RP0/CPU0:router# utility du

            160 ./schema
```

```
  104 ./lib/cerrno
  625 ./lib/mib
 2545 ./lib
 9658 ./bin
   39 ./startup
  840 ./parser
   37 ./configs
   35 ./mib
   35 ./rules
   34 ./partitions
  135 ./etc/compat
  167 ./etc
   66 ./instdb_v
  181 ./lc/bin
   33 ./lc/startup
  246 ./lc
  112 ./instdb
14006 .
```

## Related Commands

| Command | Description |
|---|---|
| utility df, on page 29 | Displays the amount of disk space available for a directory or file. |

# utility egrep

To search a file or the results of standard input using full regular expressions, use the **utility egrep** command in XR EXEC mode.

**utility egrep** {**expr** *expression* | **script** *expression-file*} [*WORD*] [**count**] [**linenum**] [**matchfile**] [**matchline**] [**nocase**] [**nofile**] [**reverse**] [**file** *search-file*]

**utility egrep** *expression* [*WORD*] [**count**] [**linenum**] [**matchfile**] [**matchline**] [**nocase**] [**nofile**] [**reverse**] [**file** *search-file*]

**utility egrep usage**

## Syntax Description

| | |
|---|---|
| **expr** *expression* | (**-e**) A regular expression. This form is used when only one expression is specified on the command line. Any names specified after this option are treated as input files. |
| **script** *expression-file* | (**-f**) A file containing a set of regular expressions, each separated by a new line. The type of the expressions is determined by the -e and -f options. This form is used when more than one expression is specified. You can specify more than one -f option.<br>The syntax of the *expression-file* argument is: [*device* **:**]/ *filename* |
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 20. |
| **count** | (Optional) (**-c**) Displays a count of selected lines. |
| **linenum** | (Optional) (**-n**) Before each output line, displays the line's line number. |
| **matchfile** | (Optional) (**-l**) (“el”) Displays only the names of files containing the selected lines. |
| **matchline** | (Optional) (**-x**) Includes only input lines selected against an entire fixed string or regular expression. |
| **nocase** | (Optional) (**-i**) Ignores uppercase and lowercase distinctions during comparisons. |
| **nofile** | (Optional) (**-h**) Displays results without a filename prefix attached to the matched lines. This option applies only when more than one file is searched. |
| **reverse** | (Optional) (**-v**) Selects only those lines that don't match the specified patterns. |
| **file** *search-file* | (Optional) The file used for the search. Replace the *search-file* argument with the device and directory path of the file. The syntax for the *search-file* argument is: [*device* **:**]/ *filename*. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If no files are specified, the keyboard input (standard input) is used.

If more than one input file is specified, then the filename is displayed before each line.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The grep utility searches files for character patterns using regular expressions and returns all lines that contain that pattern. The **utility egrep** command uses full regular expressions (expressions using the full set of alphanumeric and special characters) to match the patterns.

The results are displayed to the standard output (terminal screen).

**Note** The egrep utility options are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. For example, the **count** keyword can also be entered using the UNIX-equivalent (**-c**). To display the UNIX-equivalent syntax online, enter the **usage** keyword.

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the **utility egrep** command is used to locate the regular expression "uptime". The **linenum** keyword is also entered to display the line number before each line of output.

```
RP/0/RP0/CPU0:router# show version | utility egrep expr uptime linenum

7:router uptime is 5 days, 19 hours, 27 minutes
```

In the following example, the **utility egrep** command is used to locate a regular expression in a file. In this example, all lines with "adm" are displayed. The * character is used as a wildcard.

```
RP/0/RP0/CPU0:router# utility egrep expr Adm* nofile file disk0:/usr/passwd

adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
ssadmin:x:901218:60001:Sun StorEdge(tm) Configuration Service Agent Admin:/:/bin/false
```

In the following example, the **nocase** keyword is used to ignore the character case:

```
RP/0/RP0/CPU0:router# utility egrep expr Adm* nocase file disk0:/usr/passwd

adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
ssadmin:x:901218:60001:Sun StorEdge(tm) Configuration Service Agent Admin:/:/bin/false
```

In the following example, the **linenum** keyword is used to append the line number to the beginning of each output line:

```
RP/0/RP0/CPU0:router# utility egrep expr Adm* linenum file disk0:/usr/passwd

  5:adm:x:4:4:Admin:/var/adm:
  6:lp:x:71:8:Line Printer Admin:/usr/spool/lp:
  7:uucp:x:5:5:uucp Admin:/usr/lib/uucp:
  8:nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
  9:listen:x:37:4:Network Admin:/usr/net/nls:
 15:ssadmin:x:901218:60001:Sun StorEdge(tm) Configuration Service Agent Admin:/:/bin/false
```

## Related Commands

| Command | Description |
|---|---|
| utility fgrep, on page 9 | Searches a file for a fixed character string. |

# utility head

To copy bytes or lines at the beginning of a file or from the standard input, use the **utility head** command in XR EXEC mode.

**utilityhead**[*WORD* | [**bytes**] [**count** *number*][**file** *source*] | **usage**]

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **bytes** | (Optional) (**-c**) Copies the data in bytes from the beginning of each specified file. The default setting is to copy lines of data. |
| **count** *number* | (Optional) (**-n**) Specifies the number of lines (default) or bytes to be copied. The *number* argument is an unsigned decimal integer.<br>By default, the **utility head** command copies the first ten units (lines or bytes) of the file. Use the **count** *number* option to change the default. |
| **file** *source* | (Optional) Specifies the storage device, directory, and filename for the files.<br>If a file is not specified, the standard input is used. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If the **utility head** command is entered without keywords or arguments, the first ten lines of the file or standard output are copied.

If no file is specified, then the standard input is used.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The **utility head** command copies the beginning bytes (default) or lines of one or more files to the standard output (usually the user interface display). Use the **bytes** or **lines** keywords to copy the data based on lines or bytes. Use the **count** *number* option to specify the number of bytes or lines to copy. By default, the **utility head** command copies the first 10 lines of each file.

If more than one file is selected, an identifying header is added before the output for each file. If no file is specified, then the standard input (keyboard) is used.



**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

In the following example, the **utility head** command is used to display the first 15 lines from the output of the **show version** command:

```
RP/0/RSP0/CPU0:router# show version | utility head count 15

Tue Jul 28 06:15:44.736 DST

Cisco IOS XR Software, Version 3.9.0.14I[DT_IMAGE]
Copyright (c) 2009 by Cisco Systems, Inc.

ROM: System Bootstrap, Version 1.1(20090521:183759) [ASR9K ROMMON],

PE44_ASR-9010 uptime is 1 week, 6 days, 14 hours, 54 minutes
System image file is "bootflash:disk0/asr9k-os-mbi-3.9.0.14I/mbiasr9k-rp.vm"

cisco ASR9K Series (MPC8641D) processor with 4194304K bytes of memory.
MPC8641D processor at 1333MHz, Revision 2.2

2 Management Ethernet
12 TenGigE
40 GigabitEthernet
```

In the following example, the **utility head** command is entered with the **bytes** keyword. Only the first 15 bytes of output are displayed.

```
RP/0/RP0/CPU0:router# show version | utility head count 15 bytes

Cisco IOS XR S

RP/0/RP0/CPU0:router#
```

# utility which

To locate a program file, use the **utility which** command in XR EXEC mode.

**utility which** {[[*WORD*] | [**all**] [**fullname**] [**long** [*link*]]] **program** *program-name* | **usage**}

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **all** | (Optional) (**-a**) Displays all occurrences of the program specified by the **program** *pathname* keyword and argument. |
| **fullname** | (Optional) (**-f**) Displays the full pathname of the program file. |
| **long** [*link*] | (Optional) (**-l**) ("el") Displays the long format for each program found, and also displays link information if the file is a symlink. |
| **program** *program-name* | Specifies the name of the program file. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

None

## Command Modes

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | No modification. |

## Usage Guidelines

Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the**utility which** command is entered without keywords or arguments to display the location of the perl program:

```
RP/0/RP0/CPU0:router# utility which program perl

  /pkg/bin/perl
```

In the following example, the **utility which** command is entered with the **fullname** keyword to display the full directory path of the perl program:

```
RP/0/RP0/CPU0:router# utility which fullname program perl
```

```
  /disk0:/-base-3.8.0.1I/sbin/perl
```

In the following example, the **utility which** command is entered with the **long** keyword to display additional details about the perl program file:

```
RP/0/RP0/CPU0:router# utility which long program perl

  -rwxrwxrwx  1     0         0        19245 Jul 28 14:31 /pkg/bin/perl
```

## Related Commands

| Command | Description |
|---|---|
| utility find, on page 11 | Locates a file. |

# utility xargs

To run a program from one or more argument lists, use the **utility xargs** command in

mode.

**utility xargs** [[*WORD* | **trace**] [**program** [*program-name*] [*initial-arguments*]] | **usage**]

## Syntax Description

| | |
|---|---|
| *WORD* | (Optional) UNIX command-line option string. The maximum number of characters is 80. |
| **trace** | (Optional) (**-t**) Prints each program on standard error before executing. |
| **program** | (Optional) Specifies the name of the program and initial arguments. If a program name is not specified, then the echo utility is used. |
| *program-name* | (Optional) Specifies the name of the program. If a program name is not specified, then the echo utility is used. |
| *initial-arguments* | (Optional) Specifies the initial arguments. |
| **usage** | (Optional) Displays the UNIX options supported by this command. |

## Command Default

If no program is specified, then the echo utility is used (the input lines are displayed).

## Command Modes

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | No modification. |

## Usage Guidelines

**Note** Keywords are entered using the displayed syntax, or with UNIX-equivalent syntax. The UNIX-equivalent syntax is displayed in parentheses () in the syntax description. To display the UNIX-equivalent syntax online, enter the **usage** keyword.

## Task ID

| Task ID | Operations |
|---|---|
| universal | execute |

In the following example, the **utility xargs** command is used to display the egress lines:

```
RP/0/RP0/CPU0:router# more disk0:/usr/files | utility xargs program grep EGRESS

  disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-101
  disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-102
  disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-103
```

```
disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-104
disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-105
disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-106
disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-107
disk0:/usr/atm.cfg:service-policy output EGRESS-GigabitEthernet0-0-0-0-108
disk0:/usr/atm.cfg:service-policy output EGRESS-common
disk0:/usr/atm.cfg:service-policy output EGRESS-common
disk0:/usr/atm.cfg:service-policy output EGRESS-common
disk0:/usr/atm.cfg:service-policy output EGRESS-common
disk0:/usr/atm.cfg:service-policy output EGRESS-common
.....
```