NAME

ls − list contents of directory

SYNOPSIS

/usr/bin/ls [**-aAbcCdeEfFghHilLmnopqrRstuvVx1@**] [*file*]...

/usr/xpg4/bin/ls [**-aAbcCdeEfFghHilLmnopqrRstuvVx1@**] [*file*]...

/usr/xpg6/bin/ls [**-aAbcCdeEfFghHilLmnopqrRstuvVx1@**] [*file*]...

DESCRIPTION

For each *file* that is a directory, **ls** lists the contents of the directory. For each *file* that is an ordinary file, **ls** repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory (**.**) is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format for output directed to a terminal is multi−column with entries sorted down the columns. The **-1** option allows single column output and **-m** enables stream output format. In order to determine output formats for the **-C**, **-x**, and **-m** options, **ls** uses an environment variable, COLUMNS, to determine the number of character positions available on one output line. If this variable is not set, the **terminfo**(4) database is used to determine the number of columns, based on the environment variable, TERM. If this information cannot be obtained, 80 columns are assumed.

The mode printed when the **-e**, **-E**, **-g**, **-l**, **-n**, **-o**, **-v**, **-V**, or -@ option is in effect consists of eleven characters. The first character can be one of the following:

d The entry is a directory.

D The entry is a door.

l The entry is a symbolic link.

b The entry is a block special file.

c The entry is a character special file.

p The entry is a **FIFO** (or "named pipe") special file.

P The entry is an event port.

s The entry is an **AF_UNIX** address family socket.

– The entry is an ordinary file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of others in the user-group of the file; and the last to all others. Within each set, the three characters indicate permission to read, to write, and to execute the file as a program, respectively. For a directory, **execute** permission is interpreted to mean permission to search the directory for a specified file. The character after permissions is an ACL or extended attributes indicator. This character is an @ if extended attributes are associated with the file and the -@ option is in effect. Otherwise, this character is a plus sign (+) character if a non-trivial ACL is associated with the file or a space character if not.

ls -l (the **l**ong list) prints its output as follows for the POSIX locale:

```
-rwxrwxrwx+ 1 smith dev    10876  May 16 9:42 part2
```

Reading from right to left, you see that the current directory holds one file, named **part2**. Next, the last time that file's contents were modified was **9:42 A.M.** on **May 16**. The file contains 10,876 characters, or bytes. The owner of the file, or the user, belongs to the group **dev** (perhaps indicating ''development''), and his or her login name is **smith**. The number, in this case **1**, indicates the number of links to file **part2** (see **cp**(1)). The plus sign indicates that there is an **ACL** associated with the file. If the -@ option has been specified, the presence of extended attributes supersede the presence of an **ACL** and the plus sign is replaced with an 'at' sign (@). Finally, the dash and letters tell you that user, group, and others have permissions to read, write, and execute **part2**.

The execute (**x**) symbol here occupies the third position of the three-character sequence. A – in the third position would have indicated a denial of execution permissions.

The permissions are indicated as follows:

r The file is readable.

w The file is writable.

x The file is executable.

– The indicated permission is *not* granted.

s The **s**et-user-ID or **s**et-group-ID bit is on, and the corresponding user or group execution bit is also on.

S Undefined bit-state (the set-user-ID or set-group-id bit is on and the user or group execution bit is off). For group permissions, this applies only to non-regular files.

t The 1000 (octal) bit, or sticky bit, is on (see **chmod**(1)), and execution is on.

T The 1000 bit is turned on, and execution is off (undefined bit-state).

/usr/bin/ls

l Mandatory locking occurs during access (on a regular file, the set-group-ID bit is on and the group execution bit is off).

/usr/xpg4/bin/ls and /usr/xpg6/bin/ls

L Mandatory locking occurs during access (on a regular file, the set-group-ID bit is on and the group execution bit is off).

For user and group permissions, the third position is sometimes occupied by a character other than **x** or **-**. **s** or **S** also can occupy this position, referring to the state of the set-ID bit, whether it be the user's or the group's. The ability to assume the same ID as the user during execution is, for example, used during login when you begin as root but need to assume the identity of the user you login as.

In the case of the sequence of group permissions, **l** can occupy the third position. **l** refers to mandatory file and record locking. This permission describes a file's ability to allow other files to lock its reading or writing permissions during access.

For others permissions, the third position can be occupied by **t** or **T**. These refer to the state of the sticky bit and execution permissions.

OPTIONS

The following options are supported:

/usr/bin/ls, /usr/xpg4/bin/ls, and /usr/xpg6/bin/ls

The following options are supported for all three versions:

-a Lists all entries, including those that begin with a dot (**.**), which are normally not listed.

-A Lists all entries, including those that begin with a dot (**.**), with the exception of the working directory (**.**) and the parent directory (**..**).

-b Forces printing of non-printable characters to be in the octal *\ddd* notation.

-c Uses time of last modification of the i-node (file created, mode changed, and so forth) for sorting (**-t**) or printing (**-l** or **-n**).

-C Multi-column output with entries sorted down the columns. This is the default output format.

-d If an argument is a directory, lists only its name (not its contents). Often used with **-l** to get the status of a directory.

-e The same as **-l**, except displays time to the second, and with one format for all files regardless of age: *mmm dd hh:mm:ss yyyy*.

-E The same as **-l**, except displays time to the nanosecond and with one format for all files regardless of age: *yyyy-mm-dd hh:mm:ss.nnnnnnnnn* (ISO 8601:2000 format).

In addition, this option displays the offset from UTC in ISO 8601:2000 standard format (+*hhmm* or -*hhmm*) or no characters if the offset is indeterminable. The offset reflects the appropriate standard or alternate offset in force at the file's displayed date and time, under the current timezone.

-f Forces each argument to be interpreted as a directory and list the name found in each slot. This option turns off **-l**, **-t**, **-s**, and **-r**, and turns on **-a**. The order is the order in which entries appear in the directory.

-g The same as **-l**, except that the owner is not printed.

-h All sizes are scaled to a human readable format, for example, **14K**, **234M**, **2.7G**, or **3.0T**. Scaling is done by repetitively dividing by **1024**.

-H If an argument is a symbolic link that references a directory, this option evaluates the file information and file type of the directory that the link references, rather than those of the link itself. However, the name of the link is displayed, rather than the referenced directory.

-i For each file, prints the i-node number in the first column of the report.

-l Lists in long format, giving mode, **ACL** indication, number of links, owner, group, size in bytes, and time of last modification for each file (see above). If the file is a special file, the size field instead contains the major and minor device numbers. If the time of last modification is greater than six months ago, it is shown in the format 'month date year' for the POSIX locale. When the LC_TIME locale category is not set to the POSIX locale, a different format of the time field can be used. Files modified within six months show 'month date time'. If the file is a symbolic link, the filename is printed followed by "→" and the path name of the referenced file.

-L If an argument is a symbolic link, this option evaluates the file information and file type of the file or directory that the link references, rather than those of the link itself. However, the name of the link is displayed, rather than the referenced file or directory.

-m Streams output format. Files are listed across the page, separated by commas.

-n The same as **-l**, except that the owner's **UID** and group's **GID** numbers are printed, rather than the associated character strings.

-o The same as **-l**, except that the group is not printed.

-p Puts a slash (/) after each filename if the file is a directory.

-q Forces printing of non-printable characters in file names as the character question mark (**?**).

-r Reverses the order of sort to get reverse alphabetic or oldest first as appropriate.

-R Recursively lists subdirectories encountered.

-s Indicate the total number of file system blocks consumed by each file displayed.

-t Sorts by time stamp (latest first) instead of by name. The default is the last modification time. (See **-u** and **-c**.)

-u Uses time of last access instead of last modification for sorting (with the **-t** option) or printing (with the **-l** option).

-v The same as **-l**, except that verbose ACL information is displayed as well as the **-l** output. ACL information is displayed even if the file or directory doesn't have an ACL.

-V The same as **-l**, except that compact ACL information is displayed after the **-l** output.

The **-V** option is only applicable to file systems that support NFSv4 ACLs, such as the Solaris ZFS file system.

The format of the displayed ACL is as follows:

entry_type : *permissions* : *inheritance_flags* : *access_type*

entry_type is displayed as one of the following:

user:*username*

Additional user access for *username*.

group:*groupname*

Additional group access for group *groupname*.

owner@

File owner.

group@

File group owner.

everyone@

Everyone access, including file owner and file group owner. This is not equivalent to the POSIX other class.

The following permissions, supported by the NFSv4 ACL model, are displayed by using the **-v** or **-V** options:

read_data (r)	Permission to read the data of a file.
list_directory (r)	Permission to list the contents of a directory.
write_data (w)	Permission to modify a file's data. anywhere in the file's offset range.
add_file (w)	Permission to add a new file to a directory.
append_data (p)	The ability to modify a file's data, but only starting at EOF.
add_subdirectory (p)	Permission to create a subdirectory to a directory.
read_xattr (R)	Ability to read the extended attributes of a file.
write_xattr (A)	Ability to create extended attributes or write to the extended attribute directory.
execute (x)	Permission to execute a file.
read_attributes (a)	The ability to read basic attributes (non-ACLs) of a file.
write_attributes (W)	Permission to change the times associated with a file or directory to an arbitrary value.

delete (d)	Permission to delete a file.
delete_child (D)	Permission to delete a file within a directory.
read_acl (r)	Permission to read the ACL of a file.
write_acl (C)	Permission to write the ACL of a file.
write_owner (o)	Permission to change the owner of a file.
synchronize (s)	Permission to access file locally at server with synchronize reads and writes.
-	No permission granted

The following inheritance flags, supported by the NFSv4 ACL model, are displayed by using the **-v** or **-V** options:

file_inherit (f)	Inherit to all newly created files.
dir_inherit (d)	Inherit to all newly created directories.
inherit_only (i)	When placed on a directory, do not apply to the directory, only to newly created files and directories. This flag requires that either **file_inherit** and or **dir_inherit** is also specified.
no_propagate (n)	Indicates that ACL entries should be inherited to objects in a directory, but inheritance should stop after descending one level. This flag is dependent upon either **file_inherit** and or **dir_inherit** also being specified.
S	Successful access.
F	Failed access.
-	No permission granted.

access_type is displayed as one of the following types:

allow

Permission field that specifies allow permissions

deny

Permission field that specifies deny permissions

For example:

```
$ ls -dV /sandbox/dir.1
 drwxr-xr-x+  2 root     root           2 Jan 17 15:09 dir.1
                 user:marks:r----------:fd----:allow
                    owner@:----------:------:deny
                    owner@:rwxp---A-W-Co-:------:allow
                    group@:-w-p----------:------:deny
                    group@:r-x-----------:------:allow
                  everyone@:-w-p---A-W-Co-:------:deny
                  everyone@:r-x---a-R-c--s:------:allow
$
                              |||||||||||||:||||||+ failed access
                              |||||||||||||:|||||+--success access
                              |||||||||||||:||||+-- no propagate
                              |||||||||||||:|||+--- inherit only
                              |||||||||||||:||+---- directory inherit
                              |||||||||||||:|+----- file inherit
                              |||||||||||||
                              ||||||||||||+ sync
                              ||||||||||||+- change owner
                              ||||||||||||+-- write ACL
                              ||||||||||||+--- read ACL
                              |||||||||||+---- write extended attributes
                              |||||||||||+----- read extended attributes
                              |||||||||+------ write attributes
                              |||||||+------- read attributes
                              ||||||+-------- delete child
                              |||||+--------- delete
                              ||||+---------- append
                              |||+----------- execute
                              ||+------------ write data
                              |+------------- read data
```

-x Multi-column output with entries sorted across rather than down the page.

-1 Prints one entry per line of output.

-@ The same as **-l**, except that extended attribute information supersede **ACL** information. An @ is displayed after the file permission bits for files that have extended attributes.

/usr/bin/ls

-F Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), **FIFO**s with a trailing vertical bar (|), symbolic links with a trailing "at" sign (@), and **AF_UNIX** address family sockets with a trailing equals sign (=). Follows symlinks named as operands.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: **-C** and **-l** (ell), **-m** and **-l** (ell), **-x** and **-l** (ell), **-@** and **-l** (ell). The **-l** (ell) option overrides the other option specified in each pair.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: **-C** and **-1** (one), **-H** and **-L**, **-c** and **-u**, and **-e** and **-E**. The last option specified in each of these pairs determines the output format.

/usr/xpg4/bin/ls

-F Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), **FIFO**s with a trailing vertical bar (|), symbolic links with a trailing "at" sign (@), and **AF_UNIX** address family sockets with a trailing equals sign (=). Follows symlinks named as operands.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: **-C** and **-l** (ell), **-m** and **-l** (ell), **-x** and **-l** (ell), **-@** and **-l** (ell), **-C** and **-1** (one), **-H** and **-L**, **-c** and **-u**, and **-e** and **-E**. The last option specified in each pair determines the output format.

/usr/xpg6/bin/ls

-F Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), **FIFO**s with a trailing vertical bar (|), symbolic links with a trailing "at" sign (@), and **AF_UNIX** address family sockets with a trailing equals sign (=). Does not follow symlinks named as operands unless the **-H** or **-L** option is specified.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: **-C** and **-l** (ell), **-m** and **-l** (ell), **-x** and **-l** (ell), **-@** and **-l** (ell), **-C** and **-1** (one), **-H** and **-L**, **-c** and **-u**, and **-e** and **-E**. The last option specified in each pair determines the output format.

OPERANDS

The following operand is supported:

file A path name of a file to be written. If the file specified is not found, a diagnostic message is output on standard error.

USAGE

See **largefile**(5) for the description of the behavior of **ls** when encountering files greater than or equal to 2 Gbyte (2^31 bytes).

EXAMPLES

Example 1 Viewing File Permissions

The following example shows how to display detailed information about a file.

```
% ls -l file.1
```

```
-rw-r--r--   1 gozer    staff      206663 Mar 14 10:15 file.1
```

The permissions string above (**-rw-r--r--**) describes that the file owner has read and write permissions, the owning group has read permisisons, and others have read permissions.

The following example shows how to display detailed information about a directory.

```
% ls -ld test.dir
drwxr-xr-x   2 gozer    staff          2 Mar 14 10:17 test.dir
```

The permissions string above (**drwxr-xr-x**) describes that the directory owner has read, write, and execute permissions, the owning group has read and execute permissions, and others have read and execute permissions.

Another example of listing file permissions is as follows:

```
% ls -l file.2
-rw-rwl---   1 gozer    staff      206663 Mar 14 10:47 file.2
```

The permissions string above (**-rw-rwl---**) describes that the file owner has read and write permissions, the owning group has read and write permissions, and the file can be locked during access.

Example 2 Displaying ACL Information on Files and Directories

The following example shows how to display verbose ACL information on a ZFS file.

```
% ls -v file.1
-rw-r--r--   1 marks    staff      206663 Mar 14 10:15 file.1
    0:owner@:execute:deny
    1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
        /write_acl/write_owner:allow
    2:group@:write_data/append_data/execute:deny
    3:group@:read_data:allow
    4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
        /write_acl/write_owner:deny
    5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow
```

The following example shows how to display compact ACL information on a ZFS directory.

```
% ls -dV test.dir
drwxr-xr-x   2 marks    staff          2 Mar 14 10:17 test.dir
          owner@:--------------:------:deny
```

```
owner@:rwxp---A-W-Co-:------:allow
group@:-w-p----------:------:deny
group@:r-x-----------:------:allow
everyone@:-w-p---A-W-Co-:------:deny
everyone@:r-x---a-R-c--s:------:allow
```

The following example illustrates the **ls -v** behavior when listing ACL information on a UFS file.

```
$ ls -v file.3
-rw-r--r--   1 root     root        2703 Mar 14 10:59 file.3
    0:user::rw-
    1:group::r--                #effective:r--
    2:mask:r--
    3:other:r--
```

Example 3 Printing the Names of All Files

This command prints the names of all files in the current directory, including those that begin with a dot (**.**), which normally do not print:

```
example% ls -a
```

Example 4 Providing File Information

Another example of a command line is:

```
example% ls -aisn
```

This command provides information on **a**ll files, including those that begin with a dot (**a**), the **i**-number—the memory address of the i-node associated with the file—printed in the left-hand column (**i**); the **s**ize (in blocks) of the files, printed in the column to the right of the i-numbers (**s**); finally, the report is displayed in the **n**umeric version of the long list, printing the **UID** (instead of user name) and **GID** (instead of group name) numbers associated with the files.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

ENVIRONMENT VARIABLES

See **environ**(5) for descriptions of the following environment variables that affect the execution of **ls**: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_TIME, LC_MESSAGES, NLSPATH, and TZ.

COLUMNS Determines the user's preferred column position width for writing multiple text-column output. If this variable contains a string representing a decimal integer, the **ls** utility calculates how many path name text columns to write (see **-C**) based on the width provided. If COLUMNS is not set or is invalid, 80 is used.

The column width chosen to write the names of files in any given directory is constant. File names are not be truncated to fit into the multiple text-column output.

EXIT STATUS

0	All information was written successfully.
>**0**	An error occurred.

FILES

/etc/group	group IDs for **ls -l** and **ls -g**
/etc/passwd	user IDs for **ls -l** and **ls -o**
/usr/share/lib/terminfo/?/*	terminal information database

ATTRIBUTES

See **attributes**(5) for descriptions of the following attributes:

/usr/bin/ls

tab() box; cw(2.75i) |cw(2.75i) lw(2.75i) |lw(2.75i) ATTRIBUTE TYPEATTRIBUTE VALUE _ AvailabilitySUNWcsu _ CSIEnabled _ Interface StabilityStable

/usr/xpg4/bin/ls

tab() box; cw(2.75i) |cw(2.75i) lw(2.75i) |lw(2.75i) ATTRIBUTE TYPEATTRIBUTE VALUE _ AvailabilitySUNWxcu4 _ CSIEnabled _ Interface StabilityStandard

/usr/xpg6/bin/ls

tab() box; cw(2.75i) |cw(2.75i) lw(2.75i) |lw(2.75i) ATTRIBUTE TYPEATTRIBUTE VALUE _ AvailabilitySUNWxcu6 _ CSIEnabled _ Interface StabilityStandard

SEE ALSO

chmod(1), **cp**(1), **setfacl**(1), **terminfo**(4), **attributes**(5), **environ**(5), **fsattr**(5), **largefile**(5), **standards**(5)

NOTES

Unprintable characters in file names can confuse the columnar output options.

The total block count is incorrect if there are hard links among the files.

The sort order of **ls** output is affected by the locale and can be overridden by the LC_COLLATE environment variable. For example, if LC_COLLATE equals **C**, dot files appear first, followed by names beginning with upper-case letters, then followed by names beginning with lower-case letters. But if LC_COLLATE equals **en_US.ISO8859-1**, then leading dots as well as case are ignored in determining the sort order.