NAME

test, **[** **-** condition evaluation utility

SYNOPSIS

test *expression*
[*expression* **]**

DESCRIPTION

The **test** utility evaluates the expression and, if it evaluates to true, returns a zero (true) exit status; otherwise it returns 1 (false). If there is no expression, **test** also returns 1 (false).

All operators and flags are separate arguments to the **test** utility.

The following primaries are used to construct expression:

-b *file*	True if *file* exists and is a block special file.
-c *file*	True if *file* exists and is a character special file.
-d *file*	True if *file* exists and is a directory.
-e *file*	True if *file* exists (regardless of type).
-f *file*	True if *file* exists and is a regular file.
-g *file*	True if *file* exists and its set group ID flag is set.
-h *file*	True if *file* exists and is a symbolic link. This operator is retained for compatibility with previous versions of this program. Do not rely on its existence; use **-L** instead.
-k *file*	True if *file* exists and its sticky bit is set.
-n *string*	True if the length of *string* is nonzero.
-p *file*	True if *file* is a named pipe (FIFO).
-r *file*	True if *file* exists and is readable.
-s *file*	True if *file* exists and has a size greater than zero.

-t *file_descriptor*
: True if the file whose file descriptor number is *file_descriptor* is open and is associated with a terminal.

-u *file* True if *file* exists and its set user ID flag is set.

-w *file* True if *file* exists and is writable. True indicates only that the write flag is on. The file is not writable on a read-only file system even if this test indicates true.

-x *file* True if *file* exists and is executable. True indicates only that the execute flag is on. If *file* is a directory, true indicates that *file* can be searched.

-z *string* True if the length of *string* is zero.

-L *file* True if *file* exists and is a symbolic link.

-O *file* True if *file* exists and its owner matches the effective user id of this process.

-G *file* True if *file* exists and its group matches the effective group id of this process.

-S *file* True if *file* exists and is a socket.

file1 **-nt** *file2* True if *file1* exists and is newer than *file2*.

file1 **-ot** *file2* True if *file1* exists and is older than *file2*.

file1 **-ef** *file2* True if *file1* and *file2* exist and refer to the same file.

string True if *string* is not the null string.

s1 **=** *s2* True if the strings *s1* and *s2* are identical.

s1 **!=** *s2* True if the strings *s1* and *s2* are not identical.

s1 **<** *s2* True if string *s1* comes before *s2* based on the binary value of their characters.

s1 **>** *s2* True if string *s1* comes after *s2* based on the binary value of their characters.

n1 **-eq** *n2* True if the integers *n1* and *n2* are algebraically equal.

n1 **-ne** *n2* True if the integers *n1* and *n2* are not algebraically equal.

n1 **-gt** *n2* True if the integer *n1* is algebraically greater than the integer *n2*.

n1 **-ge** *n2* True if the integer *n1* is algebraically greater than or equal to the integer *n2*.

n1 **-lt** *n2* True if the integer *n1* is algebraically less than the integer *n2*.

n1 **-le** *n2* True if the integer *n1* is algebraically less than or equal to the integer *n2*.

If *file* is a symbolic link, **test** will fully dereference it and then evaluate the expression against the file referenced, except for the **-h** and **-L** primaries.

These primaries can be combined with the following operators:

! *expression* True if *expression* is false.

expression1 **-a** *expression2*
True if both *expression1* and *expression2* are true.

expression1 **-o** *expression2*
True if either *expression1* or *expression2* are true.

(*expression* **)** True if expression is true.

The **-a** operator has higher precedence than the **-o** operator.

Some shells may provide a builtin **test** command which is similar or identical to this utility. Consult the builtin(1) manual page.

GRAMMAR AMBIGUITY

The **test** grammar is inherently ambiguous. In order to assure a degree of consistency, the cases described in the IEEE Std 1003.2 ("POSIX.2"), section D11.2/4.62.4, standard are evaluated consistently according to the rules specified in the standards document. All other cases are subject to the ambiguity in the command semantics.

In particular, only expressions containing **-a**, **-o**, **(** or **)** can be ambiguous.

EXIT STATUS

The **test** utility exits with one of the following values:

0	expression evaluated to true.
1	expression evaluated to false or expression was missing.
>1	An error occurred.

EXAMPLES

Implement test FILE1 -nt FILE2 using only POSIX functionality:

```
test -n "$(find -L -- FILE1 -prune -newer FILE2 2>/dev/null)"
```

This can be modified using non-standard find(1) primaries like **-newerca** to compare other timestamps.

COMPATIBILITY

For compatibility with some other implementations, the **=** primary can be substituted with **==** with the same meaning.

SEE ALSO

builtin(1), expr(1), find(1), sh(1), stat(1), symlink(7)

STANDARDS

The **test** utility implements a superset of the IEEE Std 1003.2 ("POSIX.2") specification. The primaries **<**, **==**, **>**, **-ef**, **-nt**, **-ot**, **-G**, and **-O** are extensions.

HISTORY

A **test** utility appeared in Version 7 AT&T UNIX.

BUGS

Both sides are always evaluated in **-a** and **-o**. For instance, the writable status of *file* will be tested by the following command even though the former expression indicated false, which results in a gratuitous access to the file system:

```
[ -z abc -a -w file ]
```

To avoid this, write

```
[ -z abc ] && [ -w file ]
```