

**NAME**

`fpathconf`, `pathconf` - get configuration values for files

**SYNOPSIS**

```
#include <unistd.h>
```

```
long fpathconf(int fd, int name);
```

```
long pathconf(const char *path, int name);
```

**DESCRIPTION**

`fpathconf()` gets a value for the configuration option *name* for the open file descriptor *fd*.

`pathconf()` gets a value for configuration option *name* for the filename *path*.

The corresponding macros defined in `<unistd.h>` are minimum values; if an application wants to take advantage of values which may change, a call to `fpathconf()` or `pathconf()` can be made, which may yield more liberal results.

Setting *name* equal to one of the following constants returns the following configuration options:

**`_PC_LINK_MAX`**

returns the maximum number of links to the file. If *fd* or *path* refer to a directory, then the value applies to the whole directory. The corresponding macro is `_POSIX_LINK_MAX`.

**`_PC_MAX_CANON`**

returns the maximum length of a formatted input line, where *fd* or *path* must refer to a terminal. The corresponding macro is `_POSIX_MAX_CANON`.

**`_PC_MAX_INPUT`**

returns the maximum length of an input line, where *fd* or *path* must refer to a terminal. The corresponding macro is `_POSIX_MAX_INPUT`.

**`_PC_NAME_MAX`**

returns the maximum length of a filename in the directory *path* or *fd* that the process is allowed to create. The corresponding macro is `_POSIX_NAME_MAX`.

**`_PC_PATH_MAX`**

returns the maximum length of a relative pathname when *path* or *fd* is the current working directory. The corresponding macro is `_POSIX_PATH_MAX`.

**`_PC_PIPE_BUF`**

returns the size of the pipe buffer, where *fd* must refer to a pipe or FIFO and *path* must refer to a FIFO. The corresponding macro is `_POSIX_PIPE_BUF`.

**`_PC_CHOWN_RESTRICTED`**

returns nonzero if the `chown(2)` call may not be used on this file. If *fd* or *path* refer to a directory, then this applies to all files in that directory. The corresponding macro is `_POSIX_CHOWN_RESTRICTED`.

**`_PC_NO_TRUNC`**

returns nonzero if accessing filenames longer than `_POSIX_NAME_MAX` generates an error. The corresponding macro is `_POSIX_NO_TRUNC`.

**`_PC_VDISABLE`**

returns nonzero if special character processing can be disabled, where *fd* or *path* must refer to a terminal.

**RETURN VALUE**

The limit is returned, if one exists. If the system does not have a limit for the requested resource, -1 is returned, and *errno* is unchanged. If there is an error, -1 is returned, and *errno* is set to reflect the nature of the error.

**CONFORMING TO**

POSIX.1-2001.

**NOTES**

Files with name lengths longer than the value returned for *name* equal to `_PC_NAME_MAX` may exist in the given directory.

Some returned values may be huge; they are not suitable for allocating memory.

**SEE ALSO**

[getconf\(1\)](#), [open\(2\)](#), [statfs\(2\)](#), [confstr\(3\)](#), [sysconf\(3\)](#)

**COLOPHON**

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.