

**NAME**

getdtablesize - get descriptor table size

**SYNOPSIS**

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

```
int getdtablesize(void);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

```
getdtablesize():
```

Since glibc 2.12:

```
    _BSD_SOURCE ||  
    !(_POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE >= 600)
```

Before glibc 2.12:

```
    _BSD_SOURCE || _XOPEN_SOURCE >= 500 ||  
    _XOPEN_SOURCE && _XOPEN_SOURCE_EXTENDED
```

**DESCRIPTION**

**getdtablesize()** returns the maximum number of files a process can have open, one more than the largest possible value for a file descriptor.

**RETURN VALUE**

The current limit on the number of open files per process.

**ERRORS**

On Linux, **getdtablesize()** can return any of the errors described for [getrlimit\(2\)](#); see NOTES below.

**ATTRIBUTES**

**Multithreading (see [pthreads\(7\)](#))**

The **getdtablesize()** function is thread-safe.

**CONFORMING TO**

SVr4, 4.4BSD (the **getdtablesize()** function first appeared in 4.2BSD). It is not specified in POSIX.1-2001; portable applications should employ *sysconf(\_SC\_OPEN\_MAX)* instead of this call.

**NOTES**

**getdtablesize()** is implemented as a libc library function. The glibc version calls [getrlimit\(2\)](#) and returns the current **RLIMIT\_NOFILE** limit, or **OPEN\_MAX** when that fails.

**SEE ALSO**

[close\(2\)](#), [dup\(2\)](#), [getrlimit\(2\)](#), [open\(2\)](#)

**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/>.