NAME

Cwd - get pathname of current working directory

SYNOPSIS

```
use Cwd;
my $dir = getcwd;


use Cwd 'abs_path';
my $abs_path = abs_path($file);
```

DESCRIPTION

This module provides functions for determining the pathname of the current working directory. It is recommended that getcwd (or another *cwd() function) be used in *all* code to ensure portability.

By default, it exports the functions cwd(), getcwd(), fastcwd(), and fastgetcwd() (and, on Win32, getdcwd()) into the caller's namespace.

getcwd and friends

Each of these functions are called without arguments and return the absolute path of the current working directory.

getcwd

```
my $cwd = getcwd();
```

Returns the current working directory.

Exposes the POSIX function getcwd(3) or re-implements it if it's not available.

cwd

```
my $cwd = cwd();
```

The cwd() is the most natural form for the current architecture. For most systems it is identical to `pwd` (but without the trailing line terminator).

fastcwd

```
my $cwd = fastcwd();
```

A more dangerous version of getcwd(), but potentially faster.

It might conceivably chdir() you out of a directory that it can't chdir() you back into. If fastcwd encounters a problem it will return undef but will probably leave you in a different directory. For a measure of extra security, if everything appears to have worked, the fastcwd() function will check that it leaves you in the same directory that it started in. If it has changed it will `die` with the message "Unstable directory path, current directory changed unexpectedly". That should never happen.

fastgetcwd

```
  my $cwd = fastgetcwd();
```

The fastgetcwd() function is provided as a synonym for cwd().

getdcwd

```
my $cwd = getdcwd();
my $cwd = getdcwd('C:');
```

The getdcwd() function is also provided on Win32 to get the current working directory on the

specified drive, since Windows maintains a separate current working directory for each drive. If no drive is specified then the current drive is assumed.

This function simply calls the Microsoft C library _getdcwd() function.

abs_path and friends

These functions are exported only on request. They each take a single argument and return the absolute pathname for it. If no argument is given they'll use the current working directory.

abs_path

```
my $abs_path = abs_path($file);
```

Uses the same algorithm as getcwd(). Symbolic links and relative-path components ("." and "..") are resolved to return the canonical pathname, just like realpath(3).

realpath

```
my $abs_path = realpath($file);
```

A synonym for abs_path().

fast_abs_path

```
my $abs_path = fast_abs_path($file);
```

A more dangerous, but potentially faster version of abs_path.

$ENV{PWD}

If you ask to override your chdir() built-in function,

```
use Cwd qw(chdir);
```

then your PWD environment variable will be kept up to date. Note that it will only be kept up to date if all packages which use chdir import it from Cwd.

NOTES

- Since the path seperators are different on some operating systems ('/' on Unix, ':' on MacPerl, etc...) we recommend you use the File::Spec modules wherever portability is a concern.

- Actually, on Mac OS, the `getcwd()`, `fastgetcwd()` and `fastcwd()` functions are all aliases for the `cwd()` function, which, on Mac OS, calls `pwd`. Likewise, the `abs_path()` function is an alias for `fast_abs_path()`.

AUTHOR

Originally by the perl5-porters.

Maintained by Ken Williams <KWILLIAMS@cpan.org>

COPYRIGHT

SEE ALSO

File::chdir