## NAME

File::Spec::Win32 - methods for Win32 file specs

## SYNOPSIS

```
require File::Spec::Win32; # Done internally by File::Spec if needed
```

## DESCRIPTION

See File::Spec::Unix for a documentation of the methods provided there. This package overrides the implementation of these methods, not the semantics.

devnull
: Returns a string representation of the null device.

tmpdir
: Returns a string representation of the first existing directory from the following list:

```
$ENV{TMPDIR}
$ENV{TEMP}
$ENV{TMP}
SYS:/temp
C:\system\temp
C:/temp
/tmp
/
```

The SYS:/temp is preferred in Novell NetWare and the C:systemtemp for Symbian (the File::Spec::Win32 is used also for those platforms).

If running under taint mode, and if the environment variables are tainted, they are not used.

case_tolerant
: MSWin32 case-tolerance depends on *GetVolumeInformation()* `$ouFsFlags` == FS_CASE_SENSITIVE, indicating the case significance when comparing file specifications. Since XP FS_CASE_SENSITIVE is effectively disabled for the NT subsubsystem. See http://cygwin.com/ml/cygwin/2007-07/msg00891.html Default: 1

file_name_is_absolute
: As of right now, this returns 2 if the path is absolute with a volume, 1 if it's absolute with no volume, 0 otherwise.

catfile
: Concatenate one or more directory names and a filename to form a complete path ending with a filename

canonpath
: No physical check on the filesystem, but a logical cleanup of a path. On UNIX eliminated successive slashes and successive "/.". On Win32 makes

```
dir1\dir2\dir3\..\..\dir4 -> \dir\dir4 and even
dir1\dir2\dir3\...\dir4 -> \dir\dir4
```

splitpath
: ```
($volume,$directories,$file) = File::Spec->splitpath( $path );
($volume,$directories,$file) = File::Spec->splitpath( $path,
$no_file );
```

Splits a path into volume, directory, and filename portions. Assumes that the last file is a path unless the path ends in '', '.', '..' or `$no_file` is true. On Win32 this means that `$no_file` true makes this return ( `$volume`, `$path`, '' ).

Separators accepted are  and /.

Volumes can be drive letters or UNC sharenames (servershare).
```

The results can be passed to ''catpath'' to get back a path equivalent to (usually identical to) the original path.

splitdir
The opposite of *catdir()*.

```
@dirs = File::Spec->splitdir( $directories );
```

`$directories` must be only the directory portion of the path on systems that have the concept of a volume or that have path syntax that differentiates files from directories.

Unlike just splitting the directories on the separator, leading empty and trailing directory entries can be returned, because these are significant on some OSs. So,

```
File::Spec->splitdir( "/a/b/c" );
```

Yields:

```
( '', 'a', 'b', '', 'c', '' )
```

catpath
Takes volume, directory and file portions and returns an entire path. Under Unix, `$volume` is ignored, and this is just like *catfile()*. On other OSs, the `$volume` become significant.

**Note For File::Spec::Win32 Maintainers**
File::Spec::Win32 Maintainers Novell NetWare inherits its File::Spec behaviour from File::Spec::Win32.

## COPYRIGHT



## SEE ALSO

See File::Spec and File::Spec::Unix. This package overrides the implementation of these methods, not the semantics.