NAME

wcd − Wherever Change Directory

chdir for DOS and Unix

SYNOPSIS

```
wcd [options] [directory]
```

DESCRIPTION

Overview

Wcd is a command-line program to change directory fast. It saves time typing at the keyboard. One needs to type only a part of a directory name and wcd will jump to it. Wcd has a fast selection method in case of multiple matches and allows aliasing and banning of directories. Wcd also includes a full screen interactive directory tree browser with speed search.

Wcd was modeled after Norton Change Directory (NCD). NCD appeared first in *The Norton Utilities, Release 4*, for DOS in 1987, published by Peter Norton.

Wcd has been ported to different command-line shells: DOS command.com, Windows cmd.exe and PowerShell, OS/2 cmd.exe, and Unix shells such as Bourne (sh), Bourne Again (bash), Korn (ksh), Z (zsh), and C (csh) shell and others running on any operating system.

Wcd supports 8 bit character sets on all systems, and has optional support for Unicode. See section LOCALIZATION.

See section INSTALLATION how to setup wcd for personal use.

Basic use

By default (if no wcdcards are used) wcd searches for a directory with a name that begins with the typed name.

For instance this command will change to directory to the current user's `/home/user/Desktop`:

```
wcd Desk
```

When there are multiple matches, wcd will present the user a list of all matches. The user can then make a selection with a few keystrokes (most of the times only one).

Wildcards

Wcd supports following wildcards:

```
*       matches any sequence of characters (zero or more)
?       matches any character
[SET]   matches any character in the specified set,
[!SET]  or [^SET] matches any character not in the specified set.
```

A set is composed of characters or ranges; a range looks like *character hyphen character* as in `0-9` or `A-Z`. The `[0-9a-zA-Z_]` is the minimal set of characters allowed in the `[..]` pattern construct. International characters (i.e. 8 bit characters) are allowed if the system supports them. To suppress the special syntactic significance of any of `[]*?!^-\` inside or outside a `[..]` construct and match the character exactly, precede the character with a backslash (\) marker.

Using wildcards makes powerful searching possible. For instance this matched any directory name that ends with ‘‘top’’.

```
wcd *top
```

Match any directory that has contain ‘‘top’’ anywhere:

```
wcd *top*
```

Match any directory name that begins with ‘‘a’’, ‘‘b’’ or ‘‘c’’:

```
wcd [a-c]*
```

It is also possible to give a part of a directory path. Here Wcd searches for directory that begins with ‘‘Desk’’ and which path matches **me/Desk**,

```
wcd me/Desk
```

It is allowed to type any kind of expression with slashes and wildcards. E.g.:

```
wcd src*/*1?/a*2
```

Other uses

If no wildcards are used and wcd finds a perfect match, wcd will ignore all wild matches by default. This behaviour can be changed with the **−w** option.

The interactive directory tree browser can be started by using option **−g**.

```
wcd −g
```

Wcd generates a treedata file were it searches the directory. On Unix systems wcd does add links to the treedata files while scanning the disk, but does not follow them. While following links wcd could end up scanning infinite loops, or scan very large portions of a network.

Wcd can also change to directories that are not in the treedata file. E.g.:

```
wcd ..
```

If wcd found a match but can't change to the directory it tries to remove it from the default treedata file. Not from the extra treedata file. See also option **−k**.

Wcd keeps a directory stack which is stored on disk. The stack has a default size of 10 and is cyclic. See options **−z**, **−**, **+** and **=**.

In multi-user environments option **−u** can be used to change to directories of other users.

On DOS and Windows systems it does not matter if you use a slash ''/'' or a backslash ''\'' as a directory-separator.

It is possible on DOS and Windows systems to change drive and directory in one go by preceding the directory name with the drive name.

```
wcd d:games
```

Windows UNC paths

The Windows versions (Command Prompt, PowerShell, MSYS, zsh, cygwin) support Windows SMB LAN UNC paths without drive letter such as `\\servername\sharename`. Wcd for Windows Command Prompt makes use of the ''pushd'' command to automatically map a UNC path to a drive letter. In Windows PowerShell, MSYS, zsh and Cygwin UNC paths are fully supported. The current working directory can be a UNC path.

Interfaces

Wcd has three different interfaces to choose from a list of matches. The interface can be chosen at compile time.

The first interface uses plain stdin/stdout. A numbered list is printed in the terminal. The user has to choose from the list by typing a number followed by <Enter>. This interface does not provide scroll back functionality in case of a long list. The scroll back capability of the terminal/console has to be used. It is very small and portable.

The second interface is built with the conio library. It provides a builtin scroll back capability. The user is presented a list numbered with letters. Choosing from a list can be done by pressing just one letter. This interface is fast because it saves keystrokes. If possible the screen will be restored after exiting. One who prefers to type numbers can use the **−N** option.

The third interface is built with the curses library. It is similar to the conio interface. The curses version of wcd has also an additional 'graphical' interface. It lets the user select a directory via a full screen interactive directory tree browser. It has a *vim* (1) like navigation and search method. It can be activated with option **−g**.

By using the **−o** option one can always fall back to the stdin/stdout interface.

OPTIONS

−a Add current path to default treedata file.

Use this option to quickly add the current path to the default treedata file. Re-scanning the complete disk can take a long time in some cases.

−aa
Add current and all parent paths to default treedata.

−A PATH
Scan directory tree from PATH and append to the default treedata file. Examples:

```
wcd −A .
wcd −A /home −A /etc
wcd −A d: −A e: −A \\server\share
```

On Windows one can scan all shared directories of a Windows LAN server by typing something like: `wcd −A \\servername`.

See also option **−S** and **−s** and **−E**.

−b Ban current path.

Wcd places the current path in the ban file. This means that wcd ignores all matches of this directory and its sub directories.

The ban file can be edited with a text editor. Use of wildcards are supported and it is matched against absolute path.

Banned paths are not excluded from scanning the disk. To do that use option **−xf**.

−c Direct CD mode. By default wcd works as follows:

```
1. Try to find a match in the treedata file(s)
2. If no match, try to open the directory you typed.
```

In direct CD mode wcd works in reversed order.

```
1. Try to open the directory you typed.
2. If not, try to find a match in the treedata file(s).
```

−d DRIVE
Set drive for stack and go file (DOS only).

The stack file and the go-script are by default stored on drive C: if environment variable *HOME* is not set. Use this option if drive C: is a read-only drive. This option must be used in front of the stack options **−**, **+** and **=**.

−e Add current path to extra treedata file.

Use this option to quickly add the current path to the extra treedata file.

−ee Add current and all parent paths to extra treedata file.

−E PATH
Scan directory tree from PATH and append to Extra treedata file. See also options **−A** and **−S**.

−f FILE
Read treedata FILE. Do not read the default treedata file.

+f FILE
Read treedata FILE in addition to the default treedata file.

−g Graphical interface (only in version with curses interface).

Wcd starts a textual curses based ’graphical’ interface. The user can select a directory via a full-screen interactive directory tree browser. It has a *vim* (1) like navigation and search method.

If no search string is given wcd presents the whole tree which is in the default treedata file and the extra treedata files.

If a search string is given the match list is presented as a directory tree.

−ga
Graphical interface with alternative way of navigating. With this option one can't jump to unrelated directories.

−gc
Graphical interface with centered view. The selected directory stays in the middle of the screen. The centered mode can be switched on and off with key 't'.

−gd
Dump the treedata files as a tree to stdout.

−G PATH
Write go-script in directory PATH. For instance on Unix, `wcd −G PATH` will write a go-script PATH/wcd.go.

−GN
Do not create go-script. This option can be used in combination with the option **−j** if one does not want wcd to create a go-script.

−h, −−help
Print help and exit.

−i Ignore case. Dos and Windows versions of wcd ignore case default. Unix/Cygwin versions regard case by default.

+i Regard case. See also option **−i**.

−j Just go mode.

In this mode wcd will not present a list when there is more than one directory that matches the given directory. Wcd will just change to the first option. When wcd is invoked again with the same arguments it will change to the next option, and so on.

Wcd will print the directory to go to to stdout. So a different installation method can be used. One could make the following function for a POSIX compatible shell:

```
wcd ()
{
    cd "$($HOME/bin/wcd.exe −j $@)"
}
```

When you are using an old shell that doesn't support "$()" command substitution you have to use old style command substitution with backquotes.

```
wcd ()
{
    cd "`$HOME/bin/wcd.exe −j $@`"
}
```

On Windows systems, if one is running 4NT shell, one could make the following alias:

```
alias wcd `cd %@execstr[wcdwin32.exe −z 0 −j %1]`
```

This method eliminates the need of the go-script, so one can use option **−GN** in combination with **−j**.

−k Keep paths.

Keep paths in treedata when wcd can't change to them. The default behaviour of wcd is that it tries to remove paths from the treedata when wcd can't change to them. With this option this behaviour is turned off.

−K Use colors in graphical mode.

−l ALIAS
Name the current path with ALIAS. Wcd places the current path with alias ALIAS in the alias file. Aliases are case sensitive.

−m DIR
Make directory and add to treedata file.

−M DIR
Make directory and add to extra treedata file.

−n PATH
Read relative treedata file from PATH.

Do not read the default treedata file. The relative treedata file should already have been created using the wcd **+S** option. PATH may also point to a file directly.

An example: Suppose another system has been mounted to mount point `/mnt/network`:

```
wcd −n /mnt/network src
```

Wcd opens the relative treedata file in `/mnt/network/`. The file contains the paths relative from that point.

+n PATH
Read relative treedata file in addition to the default treedata file. See option **−n**.

−N Use numbers instead of letters.

Wcd with a conio or curses based interface (see section INTERFACE) presents a match list default numbered with letters. When the **−N** option is used the match list is numbered with numbers. Regardless of the **−N** option one can type a letter or numbers to make a selection from the list of matches.

−o Use stdin/stdout interface.

When for some kind of reason the conio or curses interface of wcd does not work one can fall back to the stdin/stdout interface of wcd by using the **−o** option.

−od
Dump all matches to stdout.

−q Quieter operation. Printing of the final match is suppressed.

−r DIR
Remove directory and remove from treedata file.

If the directory is empty, wcd will remove it, and try to remove it from the treedata file.

−rmtree DIR
Recursively remove directory and remove from treedata file.

Wcd will remove the directory and all its sub directories and files, and remove the directories from the treedata file.

−s (re)Scan disk from `$HOME` directory. If HOME is not defined the disk is scanned from root directory /. This is the default scanning mode.

Wcd for DOS or Windows scans the current disk from DOS root \ or from *%HOME%* if it is set.

The existing default treedata file is overwritten.

The default scan directory can be overruled with environment variable `WCDSCAN`. See section ENVIRONMENT VARIABLES.

−S PATH
Scan directory tree from PATH and overwrite the default treedata file. See also options **−A**, **−s** and **−E**. E.g. with option **−A** you can create a default treedata file of your choice. Examples:

Unix:
wcd −S /
wcd −S /home −A /etc −A /usr

DOS/Windows:
wcd −S c:/
wcd −S c: −A d: −A \\server\share

With the Windows versions one can scan all shared directories of a Windows LAN server by typing something like: `wcd −S \\servername`.

+S PATH
Scan disk from PATH and place relative paths in a relative treedata file. This file is used by the **−n** and +n options of wcd. E.g. `wcd −n PATH +src`,

−t Do not strip tmp mount dir `/tmp_mnt` (Unix only)

Wcd strips by default `/tmp_mnt/` from the match. Directory `/tmp_mnt` is used by the automounter. This behaviour can be turned off with the **−t** option.

−T Draw tree with ASCII characters. Use this option if line drawing characters are not displayed properly in your terminal.

−u USER
Scan treedata file of another user based on USER, do not scan your own default treedata file. See also section ENVIRONMENT VARIABLES for *WCDUSERSHOME*.

On Unix/Cygwin the base directory for user home directories is assumed to be `/home`. Wcd will look for `/home/USER/.treedata.wcd` and `/home/USER/.wcd/.treedata.wcd`, in that order, and read the first one that exists and is readable. On DOS/Windows the base directory for user home directories is assumed to be `\\users`, so wcd tries to read `\\users\USER\treedata.wcd` and `\\users\USER\.wcd\treedata.wcd`.

+u USER
Read default treedata file of USER in addition to your own treedata file.

−v, −−verbose
Display verbose messages. With this option wcd prints all filters, bans and excludes.

−V, −−version
Print version information and exit.

−w Wild matching only. Treat all matches as wild matches.

−x PATH
Exclude PATH from scanning.

When this option is used wcd will exclude PATH and all its subdirectories when wcd is scanning a disk. Wildcards are supported and matched against absolute paths. Option −x can be used multiple times.

```
wcd −x <path1> −x <path2> −s
```

Option −x must be used in front of any scan option (**−s**, **−S**, **+S**, **−A**, **−E**).

On DOS/Windows systems one must specify the drive letter depending on if environment variable *HOME* or *WCDHOME* is set. If *HOME* or *WCDHOME* is set one needs to specify the drive letter. An example:

```
wcd −x c:/temp −S c:
```

Otherwise do not specify drive letter.

```
wcd -x /temp -s
```

-xf FILE

Exclude all paths listed in FILE from scanning.

When this option is used wcd will exclude all paths listed in FILE and all their subdirectories when wcd is scanning a disk. Wildcards are supported and they are matched against absolute paths; one path per line. Be aware that wcd will not ignore leading or trailing blanks on a line, because they are legal characters in a directory name. Option **-xf** can be used multiple times. When one wants to exclude all banned paths from scanning one can do the following (example for wcd on unix):

```
wcd -xf ~/.ban.wcd -s
```

Wildcards are supported. For instance to exclude all your CVS directories with administrative files add a line with `*/CVS`.

Option **-xf** must be used in front of any scan option (**-s**, **-S**, **+S**, **-A**, **-E**).

-z NUMBER

Set maximum stack size to NUMBER.

The default size of the stack is 10. Stack operation can be turned off by setting the size to 0. This option must be used in front of any other stack operations (-,+,=). Otherwise the size of the stack will be set back to the default 10.

A correct command is:

```
wcd -z 50 -
```

The new stack size will be 50, wcd will go one directory back. A wrong command is:

```
wcd - -z 50
```

Wcd goes one directory back, the stack gets the default size 10. The **-z 50** is ignored.

Add this option as the first option to your wcd alias or function. E.g. for the a POSIX compatible shell this would be:

```
wcd ()
{
    wcd.exe -z 50 "$@"
    . ${WCDHOME:-${HOME}}/bin/wcd.go
}
```

-[NUMBER]

Push dir NUMBER of times. Default is one.

Go back a directory. Command `wcd -` goes one directory back. To go more directories back add a number to it. E.g. command `wcd -3`. The stack is cyclic.

+[NUMBER]

Pop dir NUMBER of times. Default is one.

Go forward a directory. Command `wcd +` goes one directory forward. To go more directories forward add a number to it. E.g. command `wcd +2`. The stack is cyclic.

=

Show stack.

Use this option if do not know anymore how many times to push or pop. The stack is printed and you can choose a number. The current place in the stack is marked with an asterisk `*`.

INSTALLATION

The current working directory of a Unix shell can only be changed by the builtin cd command. Therefore the program is always called by a function or alias. The function or alias sources a shell script (go-script)

which is generated by the wcd program. Wcd can only work after the function or alias is defined.

Another important influence on your installation is the definition of environment variables *HOME* and *WCDHOME*. See section ENVIRONMENT VARIABLES.

Install for POSIX type shells

For a POSIX shell (ksh, bash, zsh, etc.) on Unix, Linux, Cygwin, or native MSYS add the following function to the shell startup file (e.g. Bash uses `$HOME/.bashrc`):

```
wcd ()
{
    <PATH>/wcd.exe "$@"
    . ${WCDHOME:-${HOME}}/bin/wcd.go
}
```

Replace <PATH> with the location where the wcd executable has been installed. Reload the shell initialization files or start new shell.

The location of the go-script `wcd.go` differs per shell.

Wcd for DJGPP DOS bash requires a different function. The go script is not written in a directory `bin`, and if WCDHOME and HOME are both not defined the go-script is written on c:/.

```
wcd ()
{
    <PATH>/wcd.exe "$@"
    . ${WCDHOME:-${HOME:-"c:"}}/wcd.go
}
```

The WinZsh version of wcd is also a bit different. No `bin` directory.

```
wcd ()
{
    <PATH>/wcd.exe "$@"
    . ${WCDHOME:-${HOME}}/wcd.go
}
```

See section FILES for more information.

Install for C-alike shells (csh, tcsh)

Add the following alias to the shell startup file `$HOME/.cshrc` or `$HOME/.tcshrc` :

```
if ( ${?WCDHOME} ) then
    alias wcd "<PATH>/wcd.exe \!* ; source $WCDHOME/bin/wcd.go"
else
    alias wcd "<PATH>/wcd.exe \!* ; source $HOME/bin/wcd.go"
endif
```

Replace <PATH> with the location where wcd executable have been installed. Reload the shell initialization files or start new shell.

Windows Command Prompt version

In a Windows NT Command Prompt (cmd.exe) a Windows program can't change the current work directory (although a DOS program can). That is why wcd generates a batch script (`wcdgo.bat`) which must be executed in the current shell.

Windows VISTA/7

In a Windows VISTA/7 Command Prompt you may have limited access to directories. To get access to more directories you need administrator rights. You can get a Command Prompt with administrator rights if you right click on the Command Prompt icon and select *Run as administrator*.

Windows PowerShell version

Add the following function to your PowerShell user profile. The location of this profile is stored in the `$profile` variable. It is required that environment variable *HOME* or *WCDHOME* is defined.

```
function wcd
{
    <PATH>\wcdwin32psh.exe $args
    & $env:HOME\wcdgo.ps1
}
```

Replace <PATH> with the location where wcd executable have been installed. Start a new PowerShell. Wcd for PowerShell supports only the file system provider. No other providers.

OS/2 Command Prompt version

In an OS/2 Command Prompt (cmd.exe) an OS/2−program can't change the current work directory. That is why wcd generates a command script (wcdgo.cmd) which must be executed in the current shell. The script `wcd.cmd` first executes `wcdos2.exe`, which creates the `wcdgo.cmd` script. Then `wcd.cmd` executes the wcdgo.cmd script.

LOCALIZATION

LANG

The primary language is selected with the environment variable LANG. The LANG variable consists out of several parts. The first part is in small letters the language code. The second is optional and is the country code in capital letters, preceded with an underscore. There is also an optional third part: character encoding, preceded with a dot. A few examples for POSIX standard type shells:

```
export LANG=nl               Dutch
export LANG=nl_NL            Dutch, The Netherlands
export LANG=nl_BE            Dutch, Belgium
export LANG=es_ES            Spanish, Spain
export LANG=es_MX            Spanish, Mexico
export LANG=en_US.iso88591   English, USA, Latin−1 encoding
```

For a complete list of language and country codes see the gettext manual: <http://www.gnu.org/software/gettext/manual/gettext.html#Language−Codes> On Unix systems you can use to command *locale* (1) to get locale specific information.

LANGUAGE

With the *LANGUAGE* environment variable you can specify a priority list of languages, separated by colons. Wcd gives preference to *LANGUAGE* over *LANG*. For instance, first Dutch and then German: `LANGUAGE=nl:de`. You have to first enable localization, by setting *LANG* or *LC_ALL* to a value other than ''C'', before you can use a language priority list through the LANGUAGE variable. See also the gettext manual: <http://www.gnu.org/software/gettext/manual/gettext.html#The−LANGUAGE−variable>

If you select a language which is not available you will get the standard English messages.

WCDLOCALEDIR

With the environment variable *WCDLOCALEDIR* the *LOCALEDIR* used during compilation and installation of wcd can be overruled. *LOCALEDIR* is used by wcd with native language support to find the language files. The GNU default value is /usr/local/share/locale. By typing `wcd −V` wcd will print the *LOCALEDIR* that is used.

If you have installed wcd in a different directory than the default directory you may need to set the environment variable *WCDLOCALEDIR* to point to the locale directory.

An example for Windows cmd:

```
set WCDLOCALEDIR=c:/my_prefix/share/locale
```

An example for a POSIX shell:

```
export WCDLOCALEDIR=$HOME/share/locale
```

LC_COLLATE

When there are multiple directory matches wcd presents a sorted list. The sorting depends on the locale settings. If the environment LANG has been set the matches are sorted like dictionaries or phone books are sorted in that language. For instance dots and dashes are ignored, or letters e with and without accent are equal, or upper and lower case is ignored.

The sorting gives preference to environment variable *LC_COLLATE* over *LANG*. If you make *LC_COLLATE* equal to ''C'' or ''POSIX'', locale sorting is turned off. For instance if you want Dutch language, but not Dutch sorting, you can do something like this:

```
export LANG=nl_NL
export LC_COLLATE=C
```

LC_CTYPE

With regard to character encoding Wcd will give preference to variable *LC_CTYPE* over *LANG*. For instance to set character encoding to UTF−8 the following environment setting can be done.

```
export LC_CTYPE=en_US.UTF-8
```

LC_ALL

All locale environment variables that start with ''LC_'' are overruled by environment variable *LC_ALL* if it is defined. Wcd gives preference to *LC_ALL* over *LC_COLLATE* and *LC_CTYPE*.

WINDOWS CODE PAGES

There are two groups of code pages. DOS code pages (OEM) and Windows code pages (ANSI). The default encoding for Windows, when configured with Western regional settings, is ANSI CP1252. Windows programs, for instance notepad, use this default system ANSI code page. The Windows console uses by default an OEM code page (CP437 or CP850) for compatibility with DOS programs. If you use a DOS version of wcd in a Windows console it will work, because of the DOS code page. But the DOS version of wcd lacks support for long directory names and network drives on Windows. The Windows version of wcd is a native Windows program and will use the Windows system ANSI code page. So on a Western regional Windows it will use CP1252 for directory names and messages. Therefore the code page of the console has to be made equal to the system code page (changed to 1252) to make wcd for Windows work properly with special characters such as accented characters or the euro symbol. The console raster font only supports the original OEM code page installed with Windows, so you also have to change the font to true type Lucida Console to make the ANSI code page appear correctly. The Windows system code page can be changed via the Control Panel regional options. The Windows console code page is changed with the `chcp` command.

When you type `wcd -V`, the actual character encoding used by wcd is shown. Type command `chcp` to display the active code page of the Windows console.

UNICODE

Wcd has optional support for Unicode. To see if wcd was built with Unicode support type `wcd -V`. If your terminal/console and font supports it, you should see the euro symbol and Chinese characters (meaning: ''Chinese'').

Wcd has been *soft* converted to Unicode. In its core wcd handles all data as a stream of bytes. Only the lines printed to screen are on the fly converted to Unicode wide characters. Wcd fully relies on libc functions and has no UTF−8 specific code. See also <http://www.cl.cam.ac.uk/~mgk25/unicode.html>

Wcd Unicode name matching supports only binary equivalence. Matching with Unicode normalisation is not (yet) supported.

UTF−8 on Unix/Linux

In order to view UTF−8 characters your console/terminal also needs to support UTF−8. The xterm version that comes with XFree86 4.0 or higher includes UTF−8 support. To activate it, start *xterm (1)* in a UTF−8 locale and use a font with iso10646−1 encoding, for instance with

```
LC_CTYPE=en_GB.UTF-8 xterm -u8 \
-fn '-Misc-Fixed-Medium-R-SemiCondensed--13-120-75-75-C-60-ISO10646-1'
```

Modern distributions of Linux support UTF−8 by default. Other multi-byte character encodings should also

work, but that has not been tested.

UTF−16 on Windows

On Windows all the directory names on disk are encoded in UTF−16 Unicode. For programs that do not support UTF−16 the Unicode characters are translated to the active code page. For characters that are not part of the regional setting this translation is not possible and non-Unicode programs print a question mark or a wrong character instead.

Wcd with Unicode support will read the UTF−16 encoded directory names and converts them internally to UTF−8. All treedata files are encoded in UTF−8 and not compatible with the non-Unicode version of Wcd. Wcd will create a go-script encoded in UTF−8. This can only be run in Windows PowerShell. Therefore wcd with Unicode is only supported in PowerShell and not in Command Prompt. You need to set the font to True Type Lucida Console (not raster font).

UTF−8 on Cygwin

Cygwin supports Unicode since version 1.7. The Cygwin layer takes care that the Windows UTF−16 Unicode names are converted to UTF−8. So programs, like wcd, do not need to be aware of this and can operate using UTF−8 encoding as on Unix/Linux. Set character encoding to UTF−8 with the *LANG* or *LC_CTYPE* environment variable. You may need to rescan your drives. You need to set the font to True Type Lucida Console (not raster font) if you use the default Cygwin console.

FILES

If the environment variable *WCDHOME* is set wcd will use *WCDHOME* instead of *HOME*. All `*.wcd` files are text files. They can be edited with a text-editor. The Windows Command Prompt version of wcd behaves as the DOS version. The Cygwin version of wcd behaves as the Unix version.

wcd.exe

The program. In Unix shells the program is always called by a function or alias, because the current working directory of a Unix shell can only be changed by the builtin cd command. See also section INSTALLATION.

default treedata file

This is the default treedata file where wcd searches for matches. If it is not readable wcd will create a new one.

```
DOS: \treedata.wcd or %HOME%\treedata.wcd
Unix: $HOME/.treedata.wcd
```

extra treedata file

An optional extra treedata file. If it exists and is readable wcd will try to find matches in this file also.

```
DOS: \extra.wcd or %HOME%\extra.wcd
Unix: $HOME/.extra.wcd
```

ban file

In this optional file wcd places banned paths. See option **−b**. Wildcards are supported.

```
DOS: \ban.wcd or %HOME%\ban.wcd
Unix: $HOME/.ban.wcd
```

alias file

Optional file with wcd aliases. See option **−l**.

```
DOS: \alias.wcd or %HOME%\alias.wcd
Unix: $HOME/.alias.wcd
```

stack file

In this file wcd stores its stack. The drive letter can be changed with the **−d** option.

```
DOS: c:\stack.wcd or %HOME%\stack.wcd
Unix: $HOME/.stack.wcd
```

The name of the stack file can be changed with environment variable *WCDSTACKFILE*. See section

ENVIRONMENT VARIABLES.

go-script

This is the shell script which wcd.exe creates each time. It is sourced via a function or an alias. The drive letter can be changed with the **−d** option. For history reasons it is placed by default in `$HOME/bin` on Unix systems. The directory of this file can be changed with the option **−G**.

```
DOS bash: c:/wcd.go or $HOME/wcd.go
Windows Command Prompt: c:\wcdgo.bat or %HOME%\wcdgo.bat
Windows PowerShell: $env:HOME\wcdgo.ps1
WinZsh: $HOME/wcd.go
Cygwin/MSYS: $HOME/bin/wcd.go
OS/2 Command Prompt: c:\wcdgo.cmd or %HOME%\wcdgo.cmd
Unix: $HOME/bin/wcd.go
```

relative treedata file

Text file with relative paths from DIR>. See options **+S**, **−n** and **+n**.

```
DOS: <path>\rtdata.wcd
Unix: <path>/.rtdata.wcd
```

ENVIRONMENT VARIABLES

WCDHOME

Wcd uses by default environment variable *HOME* to determine where to store its files. See also section FILES.

Environment variable *WCDHOME* can be used to change the location of wcd's files. If both *HOME* and *WCDHOME* are set, *WCDHOME* will be used instead of *HOME*.

In wcd versions prior to 5.1.5 WCDHOME also changed the default scan directory. This has changed. Since version 5.1.5 WCDHOME does not change the default scan directory. See option −s. From version 5.1.5, use environment *WCDSCAN* to overrule the default scan directory.

For the Unix, Cygwin, Windows PowerShell, WinZsh and MSYS version it is required that *HOME* or *WCDHOME* is set. For the other versions of wcd the use of these variables is optional.

If HOME is set on DOS/Windows, wcd will place all its files (treedata.wcd, extra.wcd, alias.wcd, ban.wcd, wcd.go) in directory *HOME*. The behaviour of wcd is then equal to the Unix version of wcd. Wcd will scan the disk default from *HOME*. Drives will not be automatically scanned by changing to them. You need to tell wcd explicitly. E.g.:

```
wcd −S c: −A d: −A e:
```

Matching of directories is now global over all scanned drives.

Example for DOS, Windows, OS/2 Command Prompt:

```
set WCDHOME=C:\Users\erwin\wcd
```

An example for POSIX type shells:

```
export WCDHOME="$HOME/.wcd"
```

An example for Csh type shells:

```
setenv WCDHOME "$HOME/.wcd"
```

WCDSCAN

Use environment variable *WCDSCAN* to overrule the default scan directory *HOME*. Define a colon separated list (Unix) to define more than one directory. On DOS/Windows make the list semi-colon separated.

Examples for DOS, Windows, OS/2 Command Prompt:

```
set WCDSCAN=C:\Users\erwin;D:\data

set WCDSCAN=%HOMEDRIVE%%HOMEPATH%;\\projectdrive\projectX
```

An example for POSIX type shells:

```
export WCDSCAN="$HOME:/projectdisk/projectX"
```

An example for Csh type shells:

```
setenv WCDSCAN "$HOME:/projectdisk/projectX"
```

WCDFILTER

Specify filters with environment variable *WCDFILTER*. All directories that do not match the filter(s) are ignored. A list can be specified by separating filters by the shell path separator. Similar as specifying the PATH variable. The case sensitivity is mandated by the Operating system.

An example for DOS, Windows, OS/2 Command Prompt:

```
set WCDFILTER=projects;doc
```

An example for POSIX type shells:

```
export WCDFILTER="projects:doc"
```

An example for Csh type shells:

```
setenv WCDFILTER "projects:doc"
```

WCDBAN

The paths specified with environment *WCDBAN* will be banned by wcd. See also option **−b**. Specify a list of paths separated by shell PATH separator

WCDEXCLUDE

The paths specified with environment *WCDEXCLUDE* will be excluded by wcd. See also options **−x** and **−xf**. Specify a list of paths separated by shell PATH separator

An example for DOS, Windows, OS/2 Command Prompt:

```
set WCDEXCLUDE=*/windows;*/temp;*CVS
```

An example for POSIX type shells:

```
export WCDEXCLUDE="/dev:/tmp:*CVS"
```

An example for Csh type shells:

```
setenv WCDEXCLUDE "/dev:/tmp:*CVS"
```

WCDUSERSHOME

Set the base of user's home directories. On DOS/Windows the default value is `\\users`. On Unix/Cygwin the default value is `/home`. This variable is used to scan treedata files of other users. See also options **−u** an **+u**. In verbose mode wcd will print all filters, bans and excludes. See option **−v**.

WCDSTACKFILE

Wcd gives preference to *WCDSTACKFILE* over the default stack file name (see section FILES). With this variable each shell (or used terminal emulator) can have its private stack of used directories.

To use a unique time based YYYYMMDD-HHMMSS file for each opened interactive shell.

```
export WCDSTACKFILE=$HOME/.wcd/stack.$(date +%Y%m%d−%H%M%S)
```

For a stack per *xterm* (1), use the xterm WINDOWID environment variable:

```
export WCDSTACKFILE=$HOME/.wcd/stack.$WINDOWID
```

For GNU *screen* (1), to use stack per screen:

```
export WCDSTACKFILE=$HOME/.wcd/stack.$WINDOW
```

TERMINFO

If the environment variable *TERMINFO* is defined, wcd with ncurses interface checks for a local terminal definition before checking in the standard place. This is useful if terminal definitions are not on a standard place. Often used standard places are `/usr/lib/terminfo` and `/usr/share/terminfo`.

PDC_RESTORE_SCREEN

Wcd with PDCurses interface recognizes the environment variable *PDC_RESTORE_SCREEN*. If this environment variable is set, PDCurses will take a copy of the contents of the screen at the time that wcd is started; when wcd exits, the screen will be restored. An example for Windows Command Prompt:

```
set PDC_RESTORE_SCREEN=1
```

Windows allows only a small buffer to be saved. So it is not always possible to restore everything. Some garbage data may be printed in the console after wcd exists if you have set a large buffer width.

SHELL

Printing of `#!$SHELL` on the first line of the go-script for POSIX type shell or C shell is needed for 8 bit characters. Some shells otherwise think that the go-script is a binary file and will not source it. In Cygwin Bash the variable *SHELL* must be set in environtment using the `export` command, otherwise wcd can't read the variable.

BASH

Wcd for DOS bash uses `$BASH` instead of `$SHELL`, because `$SHELL` points to the DOS command shell. One may need to define `$BASH` with an `export` command, otherwise wcd can't read the variable.

SEE ALSO

sh (1), *bash* (1), *csh* (1), *ksh* (1), *zsh* (1), *locale* (1), *ncurses* (1),

AUTHORS

Wcd was written by Erwin Waterlander <waterlan@xs4all.nl>

Project homepage: <http://www.xs4all.nl/~waterlan/>

SourceForge: <http://sourceforge.net/projects/wcd/>

Freshmeat: <http://freshmeat.net/projects/wcd/>

The manual page formatting was provided by Jari Aalto <jari.aalto@cante.net>.

NCD was originally written by Brad Kingsbury for Peter Norton's ''Norton Utilities'' around 1987. See also <http://www.softpanorama.org/OFM/norton_change_directory_clones.shtml>