NAME

wcd - Wherever Change Directory
chdir for DOS and Unix

SYNOPSIS

wcd [drive:][dir] [-h] [-q] [-Q] [-b] [-l] [-c] [-w] [-v]
[-e[e]] [-E <path>] [-s] [-S <path>] [-a[a]] [-A <path>]
[-[m|M|r|rmtree] <dir>] [-u <username>] [-f <treefile>]
[-g] [-n <path>] [-i] [-t] [-d <drive>] [-z #] [-[#]] [+[#]] [=]

DESCRIPTION

Wcd. Directory changer for DOS and Unix. Another NCD-clone (Norton Change Directory) that also works under Unix.

This program jumps to a (sub)directory anywhere in the directory-tree or to directories of other users if you are on a Unix network.

Wcd generates a treedata file were it searches the directory.

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 adir
```

searches for a directory matching adir* In case of multiple matches the user can choose from a list of matching directories.

The directory to jump to can be given by only the first characters or a regular expression in the way it is used in filename matching, i.e. the wildcards *, ? and [SET] can be used.

‘*’ matches any sequence of characters (zero or more)
‘?’ matches any character
[SET] matches any character in the specified set,
[!SET] or [ˆSET] 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). [0-9a-zA-Z_] is the minimal set of characters allowed in the [..] pattern construct. Other characters are allowed (ie. 8 bit characters) if your system will support them. To suppress the special syntactic significance of any of ‘‘[]*?!ˆ-\’’, in- side or outside a [..] construct and match the character exactly, precede it with a ‘‘\’’ (backslash).

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.

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

```
wcd ..
```

It is also possible to give a part of a directory path. Eg.

```
wcd adir1/adir2
```

wcd searches for directory adir2* which path matches *adir1/adir2*

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

```
wcd 1/ad

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

In DOS it does not matter if you use a slash or a backslash as directory-separator. It is also possible under DOS to change drive and directory in one go by preceding the directory name with the drive name.

```
wcd d:games
```

If *wcd* found a match but cannot change to the directory it tries to remove it from the default treedata file. *Not from the extra treedata file.*

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

FILES

wcd.exe

The binary. Do not rename it to ’wcd’ on Unix systems. On Unix systems the binary is always executed via a function or alias.

default treedata file

DOS: \treedata.wcd or %HOME%\treedata.wcd
UNIX: $HOME/.treedata.wcd

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

extra treedata file

DOS: \extra.wcd or %HOME%\extra.wcd
UNIX: $HOME/.extra.wcd

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

ban file

DOS: \ban.wcd or %HOME%\ban.wcd
UNIX: $HOME/.ban.wcd

In this optional file wcd places banned paths. See option -b.

alias file

DOS: \alias.wcd or %HOME%\alias.wcd
UNIX: $HOME/.alias.wcd

Optional file with wcd aliases. See option -l.

stack file

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

In this file wcd stores it’s stack. The drive-letter can be changed with the -d option.

go-script

DOS BASH: c:\wcd.go or %HOME%\wcd.go
WIN32 CONSOLE: c:\wcdgo.bat or %HOME%\wcdgo.bat
WIN32 ZSH: %HOME%\wcd.go
UNIX: $HOME/bin/wcd.go

This is the shell script which wcd.exe creates each time. It has to be executed via a function or an alias. The drive-letter can be changed with the -d option.

relative treedata file

DOS: <path>\rtdata.wcd
UNIX: <path>/.rtdata.wcd

Text file with relative paths from <path>. See options +S, -n and +n.

The win32 console version of wcd behaves as the DOS version. The Cygwin version of wcd behaves as the UNIX version.

All .wcd files are ASCII text files. They can be edited with a text-editor.

If the environment variable *WCDHOME* is set wcd will use WCDHOME instead of HOME.

OPTIONS

−s (re)Scan disk from your $HOME directory.

This is recommended if you are on a large Unix server network with very much users. This is the default scanning mode. *Wcd* for DOS scans the current disk from root \ or from %HOME% if HOME is set. The existing default treedata file is overwritten.

−S <path>
Scan disk from a certain path.

If you have a small Unix system like a PC with a few users you can scan the disk from /. The existing default treedata file is overwritten.

+S <path>
Scan disk from a certain path. Make *relative* treedata file.

Scan disk from path <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

−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>
Add directory tree from <path> to default treedata.

The directory tree starting from <path> is *appended* to the default treedata file.

```
Example: wcd -A .
```

−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>
Add directory tree from <path> to Extra treedata file.

The directory tree starting from <path> is *appended* to the Extra treedata file

−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).

−w Wild matching only.

Treat all matches as wild matches.

−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 subdirectories. The match is printed in unquiet operation.

−l aLias current path. *Wcd* places the current path and the alias in the alias file. Aliases are case sensitive.

−q unQuiet operation.

With this option *wcd* prints all the matches while *wcd* is scanning the treedata files. Also banned matches are printed.

−Q Quieter operation.

Printing of the final match is suppressed.

−u <username>
Add default treedata file of other user, do not scan your own default treedata file (Unix only).

Wcd now scans the ~/.treedata.wcd of another *user* It is assumed to be /home/<username>/.treedata.wcd The default treedata file is not scanned.

+u <username>
Add default treedata file of other user (Unix only).

−f <filename>
Add another treedata file to be scanned, do not scan the default treedata file.

+f <filename>
Add another treedata file to be scanned.

−n <path>
Add relative treedata file (Unix: <path>/.rtdata.wcd, DOS: <path>\rtdata.wcd), do not scan the default treedata file. If <path> is a file, wcd will add <path> instead of <path>/.rtdata.wcd or

<path>\rtdata.wcd. See also option +S.

Example:
suppose another system has been NFS mounted to mount point /mnt/network

```
wcd -n /mnt/network src
```

Wcd now opens file /mnt/network/.rtdata.wcd The file contains the paths relative from that point.

The relative treedata file should already have been created using the wcd +S option.

+n <path>
Add another relative treedata file. See option -n.

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

+i Regard case. See also option -i.

−m <dir>
Make directory and add to treedata file.

−M <dir>
Make directory and add to extra treedata file.

−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 it's subdirectories and files, and remove the directories from the treedata file.

−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 =.

−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.

−z # Set maximum stack size.

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 option (-,+,=). 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. ’-z 50’ is ignored.

Add this option as the first option to your wcd alias or function. E.g. for the bash this would be:

```
function wcd
{
   wcd.exe -z 50 $*
   source $HOME/bin/wcd.go
}
```

−[#] Push dir [# times].

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

+[#] Pop dir [# times].

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

= Show stack.

Use this option if don’t 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 ’*’.

ENVIRONMENT

Wcd uses environment variable *HOME* to determine where to store its files. See also section FILES. Environment variable *WCDHOME* overrides HOME. If both HOME and WCDHOME are set, WCDHOME will be used instead of HOME.

For the Unix and Cygwin version it is required that HOME or WCDHOME is set. For the other versions of wcd the use of these variables is optional.

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.

Wcd with PDCurses interface recognises 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. One can set this variable e.g. in AUTOEXEC.BAT. Example:

```
   set PDC_RESTORE_SCREEN=1
```

For Cygwin this would be ’export PDC_RESTORE_SCREEN=1’.

INSTALLATION

The following sections give brief information on how to install wcd. Do you want to know more? Read file *wcd.txt* which comes with the wcd distribution.

INSTALLATION DOS VERSION

16 bit version:
Make sure that wcd.exe is in your path.

32 bit version:
Make sure that wcd.exe and cwsdpmi.exe are in your path.

DOS Bourne Again Shell
make a function like this:

```
function wcd
{
  c:/bin/wcd.exe $*
  source c:/wcd.go
}
```

File wcd.go is always written to drive c: unless other specified with the -d option.

If you use environment variable HOME make a funtion like this:

```
function wcd
{
  c:/bin/wcd.exe $*
  source $HOME/wcd.go
}
```

Z Shell

The dos bash version of wcd can also be used for the win32 port of zsh if it is used under Windows 95 or 98. It is required that enviroment variable HOME or WCDHOME is set. A function like the following must be defined. This can be done in $HOME/.zshenv

```
function wcd
{
  c:/bin/wcd.exe $*
  . $HOME/wcd.go
}
```

INSTALLATION WIN32 CONSOLE VERSION

Notice that under Windows NT console (MS-DOS prompt) a win32-program cannot change the current workdirectory (although a DOS-program can). That is why wcd generates a batchscript (c:\wcdgo.bat or %HOME%\wcdgo.bat) which must be executed in the current shell.

1) Copy wcd.bat and wcdwin32.exe somewhere in PATH.

2) Edit wcd.bat depending if you use environment variable HOME or not.

Suppose you installed wcd in c:\bin

If you don’t use environment variable HOME wcd.bat looks like:

```
 @echo off
 c:\bin\wcdwin32.exe %*
 c:\wcdgo.bat
```

If you do use environment variable HOME wcd.bat looks like:

```
 @echo off
 c:\bin\wcdwin32.exe %*
 %HOME%\wcdgo.bat
```

3) To be sure that you execute the correct 'wcd.bat' you could optionally create a macro for wcd:

```
 doskey wcd=c:\bin\wcd.bat $*
```

Notice that environment variable WCDHOME overrides HOME.

Z Shell

A win32 port of zsh has been made by Amol Deshpande (ftp://ftp.blarg.net/users/amol/zsh).

It is required that enviroment variable HOME or WCDHOME is set. A function must be defined (e.g. in $HOME/.zshenv) like this:

```
function wcd
{
   c:/bin/wcdwin32.exe $*
   . $HOME/wcd.go
}
```

INSTALLATION CYGWIN VERSION

The Cygwin version behaves exactly as the unix version: Regards case by default, same filenames, etc.

1)

Make sure that environment variable HOME is set.
Create a directory $HOME/bin

2)

copy wcd.exe to your $HOME/bin directory.

3)

Add the following function to your ˜/.bashrc file.

```
function wcd
{
   $HOME/bin/wcd.exe $*
   source $HOME/bin/wcd.go
}
```

Start a new bash

INSTALLATION UNIX VERSION

Notice that under Unix a program cannot change the current workdirectory. That is why *wcd* generates a shellscript ($HOME/bin/wcd.go) which must be executed in the current shell via a function or an alias. The following examples show the installation on a system where you don't have root privileges. Read the manpage of the shell you are using on how to define a function or an alias.

===== Korn Shell (ksh): =====

1)

copy wcd.exe to your $HOME/bin directory.

2)

Add the following function to your $HOME/.kshrc file.

```
function wcd
{
   $HOME/bin/wcd.exe $*
   . $HOME/bin/wcd.go
}
```

Start a new Korn Shell

===== C Shell (csh): =====

1)

copy wcd.exe to your $HOME/bin directory.

2)

Add the following alias to your $HOME/.cshrc file.

```
alias wcd "$HOME/bin/wcd.exe \!* ; source $HOME/bin/wcd.go"
```

Start a new C Shell

===== Bourne Again Shell (bash): =====

1)

copy wcd.exe to your $HOME/bin directory.

2)
Add the following function to your $HOME/.bashrc file.

```
function wcd
{
   $HOME/bin/wcd.exe $*
   source $HOME/bin/wcd.go
}
```

Start a new bash

===== Z Shell (zsh): =====

1)

copy wcd.exe to your $HOME/bin directory.

2)
Add the following function to your $HOME/.zshenv
file.

```
function wcd
{
  $HOME/bin/wcd.exe $*
  . $HOME/bin/wcd.go
}
```

Start a new zsh

AUTHOR

Erwin Waterlander,
waterlan@xs4all.nl
erwin.waterlander@philips.com
http://www.xs4all.nl/˜waterlan/

SEE ALSO

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