NAME

wcd - Wherever Change Directory
chdir for DOS and Unix

SYNOPSIS

wcd [drive:][dir] [-A <path>] [-a[a]] [-b] [-c] [-d <drive>]
[-E <path>] [-e[e]] [-f <treefile>] [-G <path>] [-GN] [-g[d]]
[-h] [-i] [-j] [-k] [-l] [-[m|M|r|rmtree] <dir>] [-N] [-n <path>]
[-o] [-Q] [-S <path>] [-s] [-t] [-u <username>] [-V] [-v] [-w]
[-x <path>] [-xf <file>] [-z #] [-[#]] [+[#]] [=]

DESCRIPTION

Wcd. Directory changer for DOS and Unix. Another Norton Change Directory (NCD) clone with more features.

Wcd is a 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 browser with speed search.

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

For instance:

  wcd Desk

will change to directory /home/waterlan/Desktop
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).

Wcd fully supports wildcards, i.e. *, ? and [SET].

‘*’ 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 (i.e. 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).

Using wildcards makes powerful searching possible. For instance:

  wcd *top
match any directory name that ends with "top".

  wcd *top*
match any directory that has "top" in the name.

  wcd [a-c]*
match any directory name that begins with "a", "b" or "c".

It is also possible to give a part of a directory path. E.g.:

wcd me/Desk

wcd searches for directory that begins with "Desk" and which path matches *me/Desk*

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

wcd src*/*1?/a*2

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.

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

It is also 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

The Windows console version supports Windows LAN UNC paths such as \\servername\sharename.

The interactive directory browser can be started by using option -g.

wcd -g

See option -g for more information.

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.

On Unix systems a very handy option -u can be used to change to directories of other users. See option -u.

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

wcd ..

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.* 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 =.

FILES

wcd.exe
The binary. Do not rename it to 'wcd' on Unix systems. In a Bourne-like or C shell the binary is always called by a function or alias, because the current working directory of a Bourne-like or C shell can only be changed by the builtin cd command. See also section INSTALLATION.

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. Wildcards are supported.

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 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 ~/bin on Unix systems. The directory of this file can be changed with the option -G.

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

−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 .

With the Windows console version one can scan all shared directories of a Windows LAN server by typing something like: wcd -A \\servername.

−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.

−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 <path>

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

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

−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.

−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.

−G <path>
Set directory path of go-script.

−GN Don't create go-script.

This option can be used in combination with the option *-j* if one doesn't want wcd to create a go-script.

−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 browser. It has a Vi(m) 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 graphical tree.

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

−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.

−j just go mode

In this mode wcd will not present a list when there is more than 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 bash or ksh:

```
function 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 cannot change to them. The default behaviour of wcd is that it tries to remove paths from the treedata when wcd cannot change to them. With this option this behaviour is turned off.

−l aLias current path.

Wcd places the current path and the alias in the alias file. Aliases are case sensitive.

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

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

−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.

−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.

−o Use stdin/stdout interface.

When for some kind of reason the conio or curses interface of wcd doesn't work one can fall back to the stdin/stdout interface of wcd by using the -o option.

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

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

If you have a small Unix system like a PC with a few users you can for instance scan the disk from /. With the Windows console version one can scan all shared directories of a Windows LAN server by typing something like: wcd -S \\servername.

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

−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.

−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.

−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).

−V verbose operation.

With this option *wcd* prints all filters, bans and excludes.

−v Print version info.

−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 enviroment variable HOME or WCDHOME is set. If HOME or WCDHOME is set one needs to specify the drive letter. Example:

```
wcd -x c:/temp -S c:
```

Otherwise don't 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 exlude 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 cvs administrative files add a line with:
*/CVS

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

−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 $*
  . $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 '*'.

INTERFACE

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. This interface is meant for DOS systems.

The third interface is built with the *curses* library. It is similar to the conio interface with additional sideways scrolling. The curses version of wcd has also an additional *'graphical'* interface. It lets the user select a directory via a full-screen interactive directory browser. It has a Vi(m) like navigation and search method. It can be activated with option -g. This interface is portable to DOS, Windows and Unix.

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

ENVIRONMENT

HOME and WCDHOME

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.

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. 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'.

WCDFILTER

Specify filters with environment variable WCDFILTER. All directories that don't match the filter(s) are ignored. A list can be specified by separating filters with colons (:) on Unix/Cygwin and semicolons (;) on DOS/Windows systems (Similar as specifying the PATH variable). Filters are case sensitive on Unix and case insensitive on DOS/Windows.

Example Unix: export WCDFILTER=projects:doc

Example DOS/Windows: set 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 colons on Unix/Cygwin and semicolons on DOS/Windows.

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 colons on Unix/Cygwin and semicolons on DOS/Windows.

Example Unix: export WCDEXCLUDE=/dev:/tmp:*CVS

Example DOS/Windows: set WCDEXCLUDE=*/windows;*/temp;*CVS

In verbose mode wcd will print all filters, bans and excludes. See option -V.

INSTALLATION

Note that the current working directory of a **Bourne-like or C shell** can only be changed by the builtin cd command. Therefore the binary is always called by a function or alias. The function or alias sources a shell script (go-script) which is generated by the binary wcd.exe. Wcd can only work after the function or alias is defined.

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 $*
  . 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 function like this:

```
function wcd
{
  c:/bin/wcd.exe $*
  . $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 environment 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 work directory (although a DOS-program can). That is why wcd generates a batch script (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 environment 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 file names, 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 $*
  . $HOME/bin/wcd.go
}
```

Start a new bash

INSTALLATION UNIX VERSION

The following examples show the installation on a system where you don't have root privileges. Read the man page of the shell you are using on how to define a function or an alias.

===== Bourne-like shells: =====

Korn Shell (ksh, pdksh), Bourne Again Shell (bash), Z shell (zsh), ash, ...

1)

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

2)

Add the following function to a startup file of your shell. For instance in:
$HOME/.kshrc (ksh)
$HOME/.bashrc (bash)
$HOME/.zshenv (zsh)

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

Start a new 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

AUTHOR

Erwin Waterlander,
waterlan@xs4all.nl
http://www.xs4all.nl/~waterlan/

SEE ALSO

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