I moved the non-English entries from the main page. Didn't want to lose them but I sure can't read it. AycheKay

## Contents


- 1 Telnet
  - 1.1 What version?
  - 1.2 Different commands in different versions
  - 1.3 Storing aliases / functions as macros
  - 1.4 Alternatives for ls on micro builds
    - 1.4.1 Use grep
    - 1.4.2 Use echo
    - 1.4.3 Use for
    - 1.4.4 Use wc


# Telnet

Telnet is a simple way to access the command line (shell) on your WRT54G and configure the options not accessible directly from the web. Since passwords and commands via telnet are not encrypted, SSH (Secure Shell) is generally preferable. However, on builds without an SSH service (such as micro), one must do what one can.

## What version?

What version of firmware does this require? I have a WRT54G v2.02.7, and I do not see anywhere to enable SSHD under the admin tab. I also cannot connect using putty with port 23. I can access the web interface, but I cannot get anywhere on the command line.

> blast:
> So write what is your Firmware Version. I cant understand it clear. I am with the latest version of the Firmware and it functions very proper i must admit. DD-WRT v23final from 25.12.2005, my router ist version 3.1 i also overclocked my CPU from 216MHz up to 226MHz wiht the use of Telnet and SSH.

> To first poster: I don't know if it is still an issue, but it could be that you have the wrong build. As you can see in the file versions comparison table, the *micro* build does not include the SSH daemon.
> Also, you wouldn't be able to connect to port 23 with Putty unless you change the connection method to Telnet (by default it is set to ssh).
> --HKK 12:31, 15 October 2008 (CEST)

# Different commands in different versions

Maybe someone who knows about it (i.e. development team) should include the differences in the commands included with each version. For example, I just had to find out via forum search that the *micro* build does not even include the ls command (!!!). (What's more, I haven't found some kind of directory listing either, so I have no idea what is where on the router's fs >.<).
I can understand if some commands have to be left out due to space issues, but for working with the CLI, the ls would be rather important, or it would at least be nice to have known before that it isn't present at all.
--HKK 13:01, 15 October 2008 (CEST)

# Storing aliases / functions as macros

calraith 22:19, 20 January 2010 (EDT) added the following:

Linux users can write an "expect" script to auto login and auto send alias / function commands. If you don't have tcl/tk installed, install it.

As of 20 January, 2010, the snapshot version of ExtraPuTTY can handle long aliases.

# Alternatives for ls on micro builds

As I posted on this talk page before, the micro build of DD-WRT lacks the essential `ls` command. Thus you have no way of looking around on the file system.

## Use grep

I just found a very basic substitute, but nevertheless at least you see *some*thing. The trick is to 'abuse' the `grep` command which is actually used to filter the content of files/program output for a certain search word or regular expression. By looking for a random letter (doesn't matter which one) and only letting it display the count of hits (`-c` option). By limiting the number of maximum hits per file to zero (`-m 0`) you can increase search time.
The whole command reads as follows:

```
grep -c x -m 0 *
```

**Limitations:**

- Does not differ between directories and files (you'll have to try using `cd` if it is a directory)
- Does not always work (it will not list the complete content of `/proc`, for example). I don't know where the problem lies here. It might simply be that some files are locked and `grep` blocks when it tries to access them.

If you want to, you can create an alias for `ls` this way:

```
alias ls="grep -c x -m 0 *"
```

This alias probably gets erased upon router reboot. To make it persistent, you'll have to create a startup script.

## Use echo

After poking the forum via google some more (the forum search won't allow you to search for 'ls' -- to short >.>), I also stumbled over the possibility of using `echo`. The disadvantage is that it does not enter new lines after each item. It doesn't differ between directories and files either (just like grep), but it has no problems accessing all folders (even `/proc`) and it is faster than grep.
The command is simply:

```
echo *
```

--HKK 19:00, 15 October 2008 (CEST)

## Use for

These aliases use 'for' loops to list the contents of the current directory while differentiating between directories and different file attributes.

ls
: Mimics 'ls -F' ('ls --classify'). "Append indicator (one of */=>@|) to entries"

ls1
: Mimics 'ls -F1'. Same as alias 'ls' but with one file per line.

lsa
: Mimics 'ls -Fa'. Same as alias 'ls' but includes hidden files (. files).

lsa1 *or* ls1a
: Mimics 'ls -Fa1'. Same as alias 'ls1' but includes hidden files (. files).

These will run slower than 'echo *', but I think they are useful enough. Just copy and paste into your telnet session or startup script.

```
alias ls='for i in * ; do [ -e "$i" ] && ( ( [ -d "$i" ] && echo -ne "$i/\t" ) || ( [ -h "$i" ] &
alias ls1='for i in * ; do [ -e "$i" ] && ( ( [ -d "$i" ] && echo $i/ ) || ( [ -h "$i" ] && echo
alias lsa='for i in .* * ; do [ -e "$i" ] && ( ( [ -d "$i" ] && echo -ne "$i/\t" ) || ( [ -h "$i"
alias lsa1='for i in .* * ; do [ -e "$i" ] && ( ( [ -d "$i" ] && echo $i/ ) || ( [ -h "$i" ] && e
alias ls1a=lsa1
```

*Note:* Each alias should be on a single line (there are a total of five lines in the above script). If you paste and they don't come out as a single line each, your browser is adding newline characters in the middle. Try pasting into a text editor and removing the newlines. If you're using Firefox you shouldn't have a problem.

*Note 2:* On my WRT54G v8.0, startup script does nothing. Have to paste it manually each time. You might too.

Kived 06:02, 9 November 2008 (CET)

Here is a version, call it lss, that uses a comma and period to sort directories before files.

```
alias lss='for i in .* * ; do [ -e "$i" ] && ( ( [ -d "$i" ] && echo ",  /$i" ) || ( [ -h "$i" ]
```

calraith 19:33, 2 January 2010 (EDT) contributed the following:

Here's another variation of using a for loop with results colorized based on item type (directory, symlink, character special, executable or other). It tabularizes the output, and as you might imagine, is slow but useful. :)

- usage: ls [-a] [target]

```
ls(){ d=0;[ "$1" = "-a" ]&&d=1&&shift;{ [ -z $1 ]&&a="*"; }||{ [ -d $1 ]&&a="$1/*"; }||a="$1";[ $
```

## Use wc

calraith 22:14, 4 January 2010 (EDT) contributed the following:

Use wc -c to view the size of files in bytes. This is about as close to ls -l as we'll get in micro.

```
wc -c *
```