English • <u>Deutsch</u> • <u>Español</u> • <u>Français</u> • <u>Italiano</u> • <u>???</u> • <u>Polski</u> • <u>Português</u> • <u>???????</u> • <u>Svenska</u> • <u>???(????)?</u> • <u>???(??)?</u> •

You are here: **<u>DD-WRT wiki mainpage</u> / <u>Scripting</u> /** Script Examples

Contents

Which IP addresses and hostnames are used for wireless clients?

Note: Only work if you get an IP address from DHCP

```
 # mkdir -p /tmp/www
 while [ 1 ];
  do
  wl associst | awk '{print tolower($2)}' > /tmp/assocLIST
  # echo "<meta http-equiv="refresh" content="10"><b>Hostnames and IP addresses of WLAN clients</
  while read assocLINE
   do
     dumpleases | awk '/'"$assocLINE"'/ {print "Hostname: " $1, "MAC: " $2, "IP: " $3}'
   # echo "<br>";
        done < /tmp/assocLIST     # >> /tmp/www/wlan.html
  sleep 10;
done;
```

Output:

```
Hostname: tp MAC: 01:81:18:3d:49:5e IP: 192.168.2.101
```

You can change the order of "$1, $2, $3" or cut-out:

```
... awk '{print $1,$3}'
```

Output:

```
tp 192.168.2.101
```

if you want to show this in a browser remove the *#* and use: http://routerIP/user/wlan.html

To booting on startup see Startup Scripts

How can I protect this file? I want to allow showing this only if you are logged in the web interface! Please write it here, thanks

answer:name it wlan.asp 57032956191056854646430

"How can I output the signal strength (wl rssi) at the same time by feeding MAC Addr from assoclist without typing MAC addr each time in telnet?"

Keep ISP from disconnecting due to lack of traffic

Some internet service providers will drop the connection if there is no traffic for some period of time (idle time-out). With these scripts you can prevent this.

```
#!/bin/sh

while :
do
  ping -c 5 www.example.com >/dev/null      # ping 5 packets
  sleep 300                                 # wait 5 minutes
done
```

If this does not work (providers ignores ICMP packets as traffic) use:

```
#!/bin/sh

while :
do
  wget -q -O /dev/null http://www.example.com/    # download index file
  sleep 300                                        # wait 5 minutes
done
```

To run these scripts, save to a local file, make executable with 'chmod a+x <filename>' and run with './filename'

How many connections are open for each IP address?

For each active IP address on the local network (assumed to be a 192.168.x.0 network below), this prints out the number of connections that this IP address has open to hosts on the Internet. "Connections" includes both TCP and UDP - while there are no "UDP connections", Linux maintains NAT-related information about UDP traffic which is similar to that for TCP.

Which IP addresses and hostnames are used for wirelessclients? 2

This is handy for finding that person in your LAN whose P2P software opens many hundreds of connections, making the network slow for everybody.

The last command (sort -gr) sorts the results by number of connections, with the biggest number first.

```
sed -n 's%.* src=\(192.168.[0-9.]*\).*%\1%p' /proc/net/ip_conntrack | sort | uniq -c | sort -gr
```

Signal strength

Small one line scripts to give a signal level display in the style of the Linux 'wavemon' program.

wl interface

If you are using the WRT in client mode and connecting it to another AP, you might want monitor the signal level that you are receiving at the client wrt. To do this we use the 'wl' command to access the details from the wireless driver.

```
wl rssi
```

gives the signal strength in dB and

```
wl noise
```

gives the noise level in dB.

The only problem with this is that different versions of the firmware give different outputs. Versions of DD-WRT firmware up to v23 give

```
signal is <number>
```

as the output to 'wl rssi', but version 24 just gives the number only without the "signal is " bit. This means that scripts which use the outputs from the wl command must be chosen for the different firmware versions.

In client mode

We are interested in the signal to noise figure and in monitoring it continuously, so we need a small script to do it. I also decided that it would be useful to display a small graph of the s/n figure. This is an ugly script as I had to optimise it to get it to work fast enough to be useful on the slow WRT processors :(, but this does mean that it can easily just be cut and pasted into a telnet or ssh session onto the wrt as an ugly one-liner.

Firmware v23 and before

```
while true;do wl rssi;wl noise | awk '{print $3-$6}'|awk '{printf"Signal: "$1"\t";for(;j<$1;j++)p
```

How many connections are open for each IP address?   3

Firmware v24

```
while true;do (wl rssi | sed 's/rssi is \([0-9-]*\).*/\1/';wl noise | sed 's/noise is \([0-9-]*\)
```

The output will look a bit like this:

```
Signal:  8     ========
Signal:  7     =======
Signal:  8     ========
Signal:  10    ==========
Signal:  9     =========
Signal:  11    ===========
```

To terminate, just press <ctrl> + c

When using a WRT54GL v1.1 (bought 23/May/08) with dd-wrt v24 std generic, I was able to tweak the script to the following to give meaningful SNR numbers:

```
while (true); do (wl rssi; echo " " ; wl noise) | tr -d '\n' | awk '{print $1-$2}' | awk '{printf
```

```
SNR: 53 =========================================================
SNR: 52 =========================================================
SNR: 52 =========================================================
SNR: 52 =========================================================
SNR: 52 =========================================================
```

(This was during testing with the new WRT54GL in Client Bridge mode about 3m away from my (older) access point (Netgear WGR614).)

Rockus 10:42, 24 May 2008 (CEST)

In AP mode

Using it in AP mode is more difficult as firstly, you have to supply the MAC address of the client that you want to monitor, and secondly as I havent been able to get the 'wl' command to give a meaningful noise figure so you have to guess of what you think that the noise figure is.

```
while true;do wl rssi <MAC_ADDR_OF_CLIENT> | awk '{print $3+<NOISE_GUESS_FIGURE>}'|awk '{printf"S
```

(Note : I have yet to test this on firmware v24, but if it is the same as the client mode, then the following should work -- also need to test if the noise figure works for v24)

```
while true;do wl rssi <MAC_ADDR_OF_CLIENT> | awk '{print $1+<NOISE_GUESS_FIGURE>}'|awk '{printf"S
```

You will have to replace <MAC_ADDR_OF_CLIENT> with the MAC address of the client that you wish to monitor, and replace <NOISE_GUESS_FIGURE> with a guess of what you think that the noise level is at your location. I used the figure 97 as my guess for the noise level, but it is very likely that it will be different at your location.

Having to add in the noise fudge factor will probably mean that the signal to noise figure that you get will be wrong, but the script is still useful as it will give a graph which can show you the RELATIVE strength as you move the client around the room.

(If anyone knows how to get a correct noise figure, please either add it here or in the discussion tab)

Miscellaneous tweaks

Note: SMB or WGET may be substituted for JFFS on limited flash memory devices.

Log your startup

Requires JFFS. Save your startup commands to /jffs/startup.sh

Example:

```
#!/bin/sh

cat /tmp/etc/passwd | sed s/^root:/emanymton:/ > /tmp/etc/passwd
cp /jffs/options.pptpd /tmp/pptpd/options.pptpd
rmdir /tmp/www; ln -s /jffs/www /tmp
```

Make this file executable:

```
chmod +x /jffs/startup.sh
```

Save the following line to startup:
(use administration/diagnostics, save startup)

```
/jffs/startup.sh > /tmp/startup.log 2>&1
```

Your standard and error output from the scripts will be viewable in /tmp/startup.log. /tmp is stored in volatile memory (RAM) and will be erased each startup showing only the current error or success messages. It is not recommended to use /jffs for logs due to increased flash wear. Use the echo command to label sections of your startup file and verbose command options for troubleshooting.

Change root username

```
cat /tmp/etc/passwd | sed s/^root:/emanymton:/ > /tmp/etc/passwd
```

Replace 'emanymton' with your username. Save this as a startup script.

Or a simpler way

```
sed -i -e "s/^root:/emanymton:/" /tmp/etc/passwd
```

or even

```
sed -ie "s/^root:/emanymton:/" /tmp/etc/passwd
```

In AP mode

Here -i stands for "in place" and -e introduces the sed command expression

Store user web on JFFS

Requires JFFS. Create /jffs/www.

```
mkdir /jffs/www
```

Save this as a startup script:

```
rmdir /tmp/www; ln -s /jffs/www /tmp/www
```

Place your content in /jffs/www/index.html and other files in that same directory.

The default location is http://192.168.1.1/user/index.html

Require PPTPD encryption

Requires JFFS, PPTPD. Copy /tmp/pptpd/options.pptpd to /jffs and edit.

```
cp /tmp/pptpd/options.pptpd /jffs
vi /jffs/options.pptpd
```

Insert the following:

```
mppe required
```

Save the following as a startup script:

```
cp /jffs/options.pptpd /tmp/pptpd/options.pptpd
```

Use freedns.afraid.org as DDNS

Create a Jffs partition and copy this script as */jffs/etc/config/ddns.wanup*

```
#!/bin/sh
# Made by Marc-Antoine Ruel 2006
# Idea taken from http://freedns.afraid.org/forums/viewtopic.php?p=4161

# your update URL (replace by yours)
url=http://freedns.afraid.org/dynamic/update.php?abc...
# your internet interface
inet_if=ppp0
# curl utility
curl='wget -q -O /dev/tty'
# cache file for IP address
ip_cache=/tmp/ipaddr.cache

old_ip=`cat $ip_cache`
# echo old IP = $old_ip
current_ip=`ifconfig ${inet_if} | sed '/.*inet addr:/!d;s///;s/ .*//'`
```

Change root username   6

```
# echo New IP = $current_ip

if [ "$current_ip" != "$old_ip" ]; then
  $curl $url
  echo $current_ip > $ip_cache
  echo Updated IP to $current_ip
fi
```

Don't forget to do

```
chmod +x /jffs/etc/config/ddns.wanup
```

Then you can execute it to verify that it works. A temporary file will be created in $ip_cache.

Links

<u>Startup Scripts</u>

<u>Useful Scripts</u>

<u>Basic network Script Generator</u> <u>French</u> <u>German</u> <u>Spanish</u>