So far I've learned of 3 ways to configure the DD-WRT, so that it will perform DNS for your local network. I don't claim to know which way is preferred by the developer(s) of the DD-WRT.

Warning : Whatever method you use, you might not "see" the expected results on your computer until you *reboot* the computer or *restart the network interfaces*.

On linux, for instance, simply type :

```
sudo ifdown eth0
sudo ifup eth0
```

On Windows, get to the network connections page and right-click on your adapter and choose Disable, then re-Enable it. I had unpredictable results when I just tried to renew the lease and flushdns.

Afterwards, simply ping the newly added hostname to make sure it is mapped with the right ip address.

Contents

1) Editing the /etc/hosts file

This is probably the first place one attempts getting this to work if you have a *nix background. It does work here with a *caveat*; after editing /etc/hosts, restart DNSMasq. /tmp/hosts gets recreated on boot. To make this fix persistent, write a startup script to add the new records in /tmp/hosts. Since the startup script will run after dnsmasq starts, you also need to notify dnsmasq to reload.

For example, if you populate your /etc/hosts with the startup_rc script like:

```
echo '192.168.1.50    somename' >> /etc/hosts
echo '192.168.1.51    othername otheralias' >> /etc/hosts
killall -HUP dnsmasq
```

NOTE: Previous comments suggested that SIGHUP does not cause dnsmasq to re-read the hosts file. The current web page says that it does reload hosts on SIGHUP, and testing shows it works. If you have older software the alternate solution is to kill then start dnsmasq. Previous comments also showed setting the path. I didn't need to but I also include that sample.

```
export PATH=/usr/bin:/bin:/usr/sbin
... populate /tmp/hosts ....
killall dnsmasq
dnsmasq --conf-file=/tmp/dnsmasq.conf
```

Lastly, if you want the router to append your domain on DNS request, put "expand-hosts" into the DNSMasq other options

2) Additional DNS Options (in Services)

- Before v24 it was in *Administration* > *Management*
- Before v23 SP1 Final it was in *Administration->Services->DNSMasq*

You can enter DNS info about machines in this format:

```
address=/machine_name/ip_address
```

or (added by ZC)

```
dhcp-host=machine_name,ip_address
```

Multiple hostnames can be used with one IP address by adding extra [/hostname] sections.

For example:

```
address=/zinc/192.168.1.30
address=/zirconium/zr/192.168.1.31
```

or (added by ZC)

```
dhcp-host=zinc,192.168.1.30
```

Or fully qualified:

```
address=/zinc.your_domain.com/192.168.1.30
```

(edit by ZC): An additional option is to have DNSMasq read an ADDITIONAL hosts file (for those who use the hosts method of ad blocking) using this option from the DNSMasq documentation:

```
# If you don't want dnsmasq to read /etc/hosts, uncomment the following line.
#no-hosts
# or if you want it to read another file, as well as /etc/hosts, use this.
#addn-hosts=/etc/banner_add_hosts
```

To have DNSMasq automatically append the domain name when a request is served, add the "expand-hosts" option and ensure the domain name is set either in the additional options or in the DD-WRT web interface.

When using "expand-hosts", DNS lookups for "hostname.your_domain.com" will fail unless "your_domain.com" is included as part of the domain in the "address" option.

```
domain=your_domain.com
expand-hosts
address=/zirconium.your_domain.com/zr.your_domain.com/192.168.1.31
```

To achieve reverse DNS lookups, use "ptr-record":

```
address=/zirconium.your_domain.com/192.168.1.31
ptr-record=zirconium,192.168.1.31
```

(edit by ZC): If you use the "dhcp-host" method with "expand-hosts", the domain will be appended

The official DNSMasq documentation is at http://www.thekelleys.org.uk/dnsmasq/doc.html. There is a sample configuration file at http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq.conf.example.

-comment- for v23, the trailing slash, as seen in the examples, caused it not to work. As soon as it was removed, the errors seen when re-starting dnsmasq (in ssh) went away and the service started resolving hostnames

-comment - for v24sp1, the above syntax is incorrect. The following works correctly...

```
address=/host.example.net/10.1.2.30
ptr-record=30.2.1.10.in-addr.arpa,"host.example.net"
```

3) Creating static leases

Static leases from Administration->Services->DHCP Server

You just hit "add", and type them in. You need to "save" settings at the bottom of the page (Duh). The thing I don't like about this method is:

1) If you do NOT want to tie MAC addresses with IPs then you have to use a fake MAC address on this form, because the info doesn't save without a MAC address

2) If you need to enter many of these, it gets tedious

(edit by ZC): An alternate method of static leases is to use the "dhcp-host" method for addresses above, but add MAC addresses for those that need it, e.g.:

```
# This entry is simply a static DNS address, great for mapping print servers, etc to names
dhcp-host=zinc,192.168.1.30
# This entry assigns the given IP address to the MAC address for static IP addresses
# Note that the IP address listed does NOT have to be in the DHCP range given, just on the same s
dhcp-host=11:22:33:44:55:66,zinc,192.168.1.30,infinite
```

4) dnsmasq wrapper

This solution works for me, as i like to have a larger /etc/hosts file and therefore the startup command solution is not ideal. By the DD-WRT overwrites the host file, when you static leases.

(edit by ZC): Option "addn-hosts" works well here as well for using an additional, large hosts file

What it does:

1) The Script is a wrapper around dnsmasq with a little help of mount -o bind.

2) Once it is hooked and DD-WRT wants to restart dnsmasq the script is executed instead, and adds some lines to the generated /etc/hosts afterwards it executes the real dnsmasq. Finally it reinstalls the wrapper.

You have to place the script somewhere and put it in to list of commands executed at bootup. Then adjust the variables at the setup section. Let dnsmasq and tmp untouched. The hostfile is an file in /etc/hosts format. But the generation of /etc/hosts could be adjusted as you like. like. Tinker a bit in the Section "$0" = "$dnsmasq".

To do the magic i added dumb versions of readlink and pgrep to the script. They are not fast and could be buggy.

```
#!/bin/sh
#- setup
        dnsmasq=/usr/sbin/dnsmasq
        wrapper=/jffs/local/bin/dnsmasq-wrapper.sh
        #wrapper=/tmp/root/dnsmasq-wrapper.sh
        hostfile=/jffs/local/etc/hosts
        #hostfile=/tmp/root/hosts
        tmp=/tmp/dnsmasq-wrapper.tmp.sh


#- dirty version of readlink for dd-wrt
readlink() {
        if [ -n "$1" ]; then
                 ls -l "$1" | sed 's/^.* -> \(.*\)$/\1/g' | grep -v lrwxrwxrwx
        fi
}

#- dirty version pgrep; -f -v not implemented
pgrep() {
        local i
        local exe
        local pid
        pid=
        for i in /proc/*/exe; do
                exe="`readlink "$i"`"
                if [ "$exe" == "$1" ]; then
                        i="${i%/*}"
                        pid="${i##*/}"
                        echo -n "$pid "
                fi
        done
        if [ -n "$pid" ]; then
                echo
        fi
}

#- debug
        #set -x


if [ "$0" = "$dnsmasq" ]; then
        if [ -f "$hostfile" ]; then
                cat "$hostfile" >> /etc/hosts
        fi

cat > "$tmp" <<EOF
                while umount "$dnsmasq"; do :; done
                $0 $@ &
                sleep 1
                while umount "$dnsmasq"; do :; done
                #mount -o bind "$0" "$dnsmasq"
                mount -o bind "$wrapper" "$dnsmasq"
                rm -f "$tmp"
EOF
        exec $SHELL "$tmp"
else
        echo "$0: wrapper installed."
        while umount "$dnsmasq"; do :; done
        #mount -o bind "$0" "$dnsmasq"
        mount -o bind "$wrapper" "$dnsmasq"
        pid="`pgrep "$dnsmasq"`"        #- pid von dnsmasq feststellen
        if [ -n "$pid" ]; then
                pid="${pid%% *}"        #- nehme erstes elt aus pid
                cmdline="`tr '\000' ' ' < /proc/$pid/cmdline`"
                kill $pid
                exec $cmdline
        fi
fi
```

5) DNSMasq and DNS name resolution thru VPN tunnel

If you are using DNSMasq for DNS and also use a VPN client on your Windows PC's (say to connect to work), you may find that your DNS name resolution through your VPN tunnel may stop working. This is because the DNS requests are being sent to DD-WRT instead of the VPN tunnel, due to the binding order of the remote access connection.

See this Microsoft technote for how to fix this:
http://support.microsoft.com/default.aspx?scid=kb;en-us;311218

The technote did not work for me as written because my VPN client had its own network adapter and did not use \Device\NdisWanIp. However moving the GUID of the VPN network adapter (mine was listed with ipconfig /all) to the top of the **Bind** value instead of \Device\NdisWanIp corrected all my problems with VPN name resolution.

Thanks to dmulk for providing the technote link.

6) Custom dnsmasq started from jffs2 or mmc

One easy way to use a custom dnsmasq is to add a symlink to dnsmasq somewhere in your jffs or mmc filesystem with a different name, eg: mydnsmasq. Then you write a script that basically copies the original config, adds the customizations and then does a "killall dnsmasq" and starts you custom one using the "mydnsmasq" symlink. The process will have a different name and will use a different config file so it won't be catched by the check_ps and even if the real dnsmasq is started it will abort right away because the port 53 is in use by your custom one.