

**WARNING** : This page is OUTDATED. Use the following link : [Netgear\\_WGR614v8/L/WW](#).

## WGR614 V8 / WGR614L

**Firmware Notes** The basic hardware for the WGR614 V8 and WGR614L is identical and for the scope of this article the two device names can be used interchangeably. Please note that to load firmware on this device it must be first recompiled in Linux! If you do not do this you will brick your router with the DD-WRT firmware as it needs to be recompiled into the format necessary for the Netgear WGR614 V8. Hence the funky extension (.CHK).

**Quick start** To install DD-WRT on a WGR614 v8 / WGR614L with the original firmware installed, first download the 'Special WGR614 v8/L image for initial flashing' from the [dd-wrt database](#) - It is a .chk file.

Flash the 'dd-wrt.v24\_mini-wgr614v8-L.chk' file, making sure to follow these [instructions](#) carefully.

You now have a fully functioning router, running a 'mini build' of DD-WRT and can install one of the 'Generic' builds if you like. The 'NoKaid' build is recommended.

Conspiritech says: Use eko firmware. I used 11100 [go to [www.dd-wrt.com](#), click downloads, follow others/eko/V24\_TNG\_OBSOLETE/svn11100 and get dd-wrt.v24-11100\_WGR614v8-L\_mini.chk] with .chk extension of course. For more info on this and other releases find eko in forums here or on [myopenrouter.com](#). Flash from web interface. Reset to factory defaults after flashing and then reboot it again just to be sure, before attempting settings. I am not even going to attempt to get a .bin or a standard on there given all the remaining funkiness in the reports about this router.

Procedure 1 on this page [Recover from a Bad Flash](#) also works for wgr614l v8. I only killed this one once, by not defaulting after flashing, the interface slows to a crawl and it crashes after a few reboots. This process defaults the machine and you are back to your current firmware. Still pings in this condition.

**Basic Guide to Flashing (Must Use .CHK Version of The Firmware)** First off if you have the WGR614L you can download this mini firmware and flash it as its already to go. [Modded Mini Firmware Ready For Installation](#) If you do not want to use that firmware you will need to download the [Netgear toolchain and source](#) for your router and rebuild the latest DD-WRT source code (available below at svn) into a working firmware. To download svn links in Windows you have to install [TortoiseSVN](#) or an equivalent.

```
svn co svn://svn.dd-wrt.com/DD-WRT
```

To install the Standard version of DD-WRT you will need to recompile it using these instructions (NOTE: Linux is required for this procedure!)

WGR614L [Firmware Image Creation](#)

**Text below taken from WGR614L Firmware Image Creation @ myopenrouter.com**

Some of you have already noticed that WGR614L firmware uses a .chk extension. Let us go through the steps that is needed to create a firmware image in .chk format.

## Netgear\_WGR614\_v8.0\_&\_WGR614L

Most of the Broadcom hardware uses a TRX style firmware. Netgear's WGR614L Linux based open source router's .chk is also based on the same TRX format - but does add the following interesting improvements

First, the kernel is compressed using LZMA which achieves a better compression than the usual gzip or bzip2 compression. This improvement does need an addition to the CFE bootloader to support decompression of LZMA encoded kernel

Second, this format adds a header to trx format, that has the checksum, length, and a compatible board identifier. These checksum and length information makes the job ensuring image integrity much easier and since these checksums are stored in the flash and also checked at each system boot, boot-loader can also identify if the flashed image gets corrupted.

So to pack a image in the .chk format you will need the following files at hand These files are generated when you compile the downloaded Linux opensource codes for WGR614L

1. Download the source code the Netgear WGR614L
2. Linux Kernel - vmlinux -usually compiled at the top directory of the Linux kernel source.
3. compatible.txt containing the boardid of the netgear boards this fimeware is targeted for.
4. Root File System image - target.squashfs - usually a LZMA Squashfs file system.
5. Make sure you have added the bcm5354/tools directory into the PATH

```
$ export PATH=$PATH:${SRCDIR}/bcm5354/tools
```

And you will need to follow these steps

1. Trim vmlinux by removing unnecessary sections in the object code

```
$ mipsel-linux-strip vmlinux
```

2. Convert to Binary format from the ELF format

```
$ mipsel-linux-objcopy -O binary -g vmlinux vmlinux.bin
```

3. Compress using lzma

```
$ ../../tools/lzma e vmlinux.bin vmlinux.lzma
```

4. Create the trx image

```
$ trx -o linux.trx vmlinux.lzma target.squashfs
```

5. Append the ".chk" header.

```
$ touch emptyfile; packet -k linux.trx -f emptyfile -b compatible.txt -ok kernel_image \
```

```
-oall kernel_rootfs_image -or rootfs_image -i emptyfile; rm -f emptyfile
```

And this will generate a new file "kernel\_rootfs\_image.chk" which can be uploaded to WGR614L using its Web UI or if you got the serial console you can also use the tftpd upgrade from the CFE firmware.

The format of a chk header is given below.

## Netgear\_WGR614\_v8.0\_&\_WGR614L

Type/Variable Size (bytes)

Unsigned long magic

4

Unsigned header\_len;

4

unsigned char reserved[8];

8

unsigned long kernel\_chksum;

4

unsigned long rootfs\_chksum;

4

unsigned long kernel\_len;

4

unsigned long rootfs\_len;

4

unsigned long image\_chksum;

4

unsigned long header\_chksum;

4

BOARD ID < 64 digits

variable length (max 64 bytes)

Read More In: [WGR614L Open Source Router Setup / Configuration Open Source Projects and Firmware](#)

**[Solution for TRX errors Solution for TRX Errors on NETGEAR Builds](#)**

**Modded Firmware Installation Guide** DD-WRT Firmware Installation Guide

**If You Brick It!** Tftp back into it. Once you get in, flash the original OEM for the device you have firmware as this is for recovery only! Do not flash any other firmware! After you bring it back from the dead you can start playing with firmware again but not now!

**So, You Killed It Twice!** Now is the time for JTAG It is highly recommended that with this particular router that you do not attempt to solder in a pin header for the JTAG. The heat from the iron will destroy the surrounding traces. Best bet is to solder the JTAG from the underside of the circuit board where the least damage can occur and to put the leads directly in the board itself after tining the wires and applying flux.