# Building ROOter Images

To create a OpenWrt build system that can be used to create ROOter images you need a computer that is running Linux.

This can be a Virtual Machine running in Windows or Linux running on a computer. The recommended Linux distros to use for this are *Linux Mint, Ubuntu 18.04 or Debian*. ROOter images are made using Linux Mint 20. This computer must have Internet access in order to download the files needed to create an OpenWrt build system.

Using the File Manager of your Linux distro, create a folder that will contain the build system. It is possible to have multiple build systems resident in this folder if you want to build images with different OpenWrt versions.

We will name this folder **OpenWrt** but it can be called anything.

Using the File Manager go to the **OpenWrt** folder and open a Terminal session. How this is done depends on the distro used. In Linux Mint you right click and chose *Open in Terminal*.

The first thing that needs to be done is to add all the extra packages to your distro that are required when building an image. This is done from the Terminal by entering the following commands.

For Linux Mint

```
sudo apt-get install build-essential subversion git-core libncurses5-dev zlib1g-dev quilt
sudo apt-get install gawk flex quilt libssl-dev xsltproc libxml-parser-perl jshon
```

For Ubuntu

```
sudo apt -y install build-essential libncurses5-dev python unzip gawk git curl jshon
```

For Debian 11

```
sudo apt install build-essential bash binutils bzip2 flex git-core g++ gcc util-linux gawk
sudo apt install help2man intltool libelf-dev zlib1g-dev make libglvnd-dev pkg-config
sudo apt install libncurses5-dev libssl-dev patch perl-modules git ncurses-dev libz-dev
sudo apt install python2-dev wget gettext xsltproc zlib1g-dev zip unzip libssl-dev
sudo apt install python2 python3-dev libelf-dev subversion gettext gawk wget curl
sudo apt install rsync perl unrar rar jshon
```

Once you have the required packages installed you can create the build system. At the Terminal enter the following lines, waiting for each to complete before entering the next one.

This will take a bit of time and requires an Internet connection.

First clone the build system from GitHub into a folder named *rooter19076*. This folder name can be changed to anything by changing the name in the command.

```
git clone https://github.com/ofmodemsandmen/RooterSource rooter19076
```

This will take some time as it must download the build system and set it up. Then move into the *rooter19076* folder and update all the build packages.

```
cd rooter19076
mkdir -p ./images
```

Once this has completed you can close the Terminal session as the build system is ready to use.

## The Automated Build System

If you are building your own ROOter images there is automated build script available that will build images for a predefined number of routers using just one line in the Terminal.

You can also use the predefined config files of a router to create a custom image with extra packages not found in a standard ROOter. These custom images are created from the template config file for the router so all that is needed is to add the extra packages of your choice. This custom config file in no way will change the template config file so it is safe to make changes to your custom config file.

If you set up your Linux build computer using the above list of extra packages you can skip the next section. If you are using an existing build system you need to install one more package.

In order to use this automated build system it is necessary to install another package on your Linux computer. This is a package called **jshon** and can be installed from the Software Manager or from the Terminal command line.

To install from the Terminal enter :

```
sudo  apt-get install jshon
```

The web site for this package is **http://kmkeen.com/jshon/**

## Using the Build Script

This script is run from the Terminal as follows.

To see the help and a list of routers that are supported by the automated build system enter :

```
./build
```

This will output information about using the build script and a list of supported routers.

*Usage : <router name> [flag config name]*

*<router name> is one of the names in the following list*

*[flag config name] are optional for working with custom config file for this router*
*[flag] : -e  create or edit a custom config file*
*-eb create or edit a custom config file and then build the image*
*-b  build an image using the custom config file*
*[config name] : name given to a custom config file for this router*
*if this config file does not exist then the template*
*config file will be used to create it*

*Supported Models are :*

| | | | |
|---|---|---|---|
| *ALIX-2D13* | *APU2C4* | *ARCHERA7V5* | *ARCHERC7V2* |
| *ARCHERC7V4* | *ARCHERC7V5* | *ARCHERC20V1* | *AR150* |
| *AR300LITE* | *AR750* | *AR750S* | *B1300* |
| *GLMIFI* | *MR3020V3* | *MYNET-N600* | *MYNET-N750* |
| *R6220* | *R7800* | *RASPPI* | *RASPPI2* |
| *RASPPI3* | *RBM11G* | *RBM33G* | *RT-AC58U* |
| *SMARTBOXPRO* | *TURBOPLUS* | *TURBOMODPLUS* | *WE826Q* |
| *WE826T* | *WE826WD* | *WG1602* | *WG209* |
| *WG3526* | *WRT1900AC* | *WRT1900ACS* | *WRT3200ACM* |
| *WRT32X* | *X750* | *X86-64* | *XMIFI3PRO* |

To build a standard ROOter image for a router you enter :

**./build *modelname***

where ***modelname*** is one of the names in the above list. The name can be in either upper or lower case as it makes no difference.

This will build all the images needed for this router and will add the extra files needed to perform the flash. The archive file will be placed in the **images** folder.

Two sets of images are included in the archive file, images **with Load Balancing** and images **without Load Balancing**. Not all routers will have both sets of images as those with 8meg of flash do not have enough space and some have it by default.

Those images that have **-full** in their name are the ones with Load Balancing. This allows you to chose if you want Load Balancing or not.

## Creating Custom Images

If you wish to create a custom image for a router that contains extra packages not found in a standard ROOter you need to create a **custom config file** for this router. Each router can have as many custom config files as you like as long as they have different names.

Custom config files are created for a specific router and, upon creation, will contain the config file for the standard ROOter image for this router. To create and edit or edit an existing custom config file enter :

### ./build *modelname -e customname*

*where* ***modelname*** is the name of a router from the above list and **customname** is the name of your custom config file. All changes made to the custom config file will have no effect on the standard config file for this router. The changes are only in the custom config file.

To create a custom config file for an **Archer C7v5** named **custom-c7v5** you would enter :

### ./build archerc7v5 -e custom-c7v5

If the config file named **custom-c7v5** does not exist it will be created from the standard ROOter config file for the **Archer C7v5**. Changes are made to the config file by using the normal **OpenWrt Configuration program**, which is invoked normally by running **make menuconfig**.

**Important Note** : when making a custom image you must select the desired ROOter package for this image. That is either **ext-rooter-lite** or **ext-rooter16**. You can then add any extra packages you like. Failing to do this will create an image without ROOter in it.

The above command will only allow you to create and edit the custom config file. If you wish to create/edit a custom config file and then build an image from it you would enter :

### ./build archerc7v5 -eb custom-c7v5

If you wish to just build an image from an existing custom config file enter :

**./build archerc7v5 -b custom-c7v5**

If the custom config file does not exist it will be created first and populated from the standard ROOter config file for this router.

All images created by the build system will be placed in a Zip archive in the ***/images*** folder.