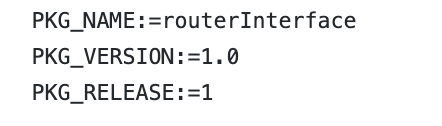
OpenWrt Cross Compilation Guide

This guide will go through the cross-compilation of projects more complicated than a Hello World.

**NOTE: Please read and follow the “**[Setting Up OpenWRT and Running a \_hello world](https://github.com/crumbj/Scenic-Routing/blob/main/Routing%20Protocol/OpenWrt%20Documentation/Setting%20Up%20OpenWRT%20and%20Running%20a%20_hello%20world_.docx)**” document before this guide. It will be referred to as the “hello world documentation”**

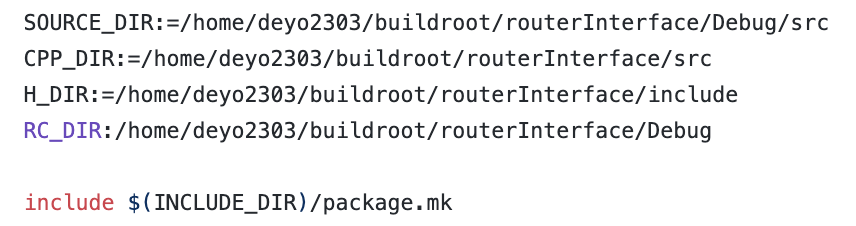
# Makefile Debunked!

This part of the makefile describes the name, version, and release of your package, which is then used to name your .ipk file. This name can be changed anytime to signify a new release or version. When you install your .ipk on the OpenWrt router, the PKG\_NAME will be what you type in the command line to run your program!



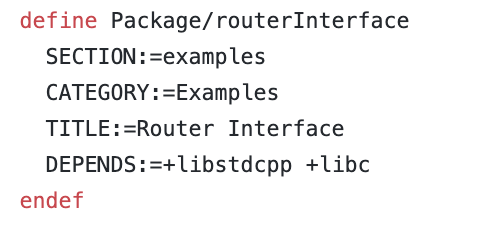
These variables are used in other places in the makefile. They are all directories from the project to cross compile:

* SOURCE\_DIR: the source directory within the debug folder where all the of the executable files are located.
* CPP\_DIR: the source directory where the .cpp files are located to be compiled into executable files in the SOURCE\_DIR location
* H\_DIR: the directory containing all of the .h to be compiled into the executable files in the SOURCE\_DIR location.
* RC\_DIR: is the directory used to get the Router Configuration files.

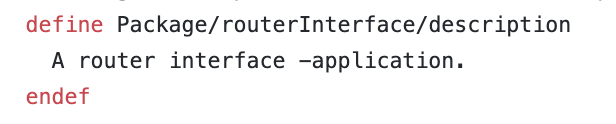


This section is the first part of the package definition. The SECTION, CATEGORY, and TITLE variables determine where the package is found in the configuration menu when you run the command “make menuconfig”.

The DEPENDS definition says that the package has the libraries libstdc++ and libc as dependencies.

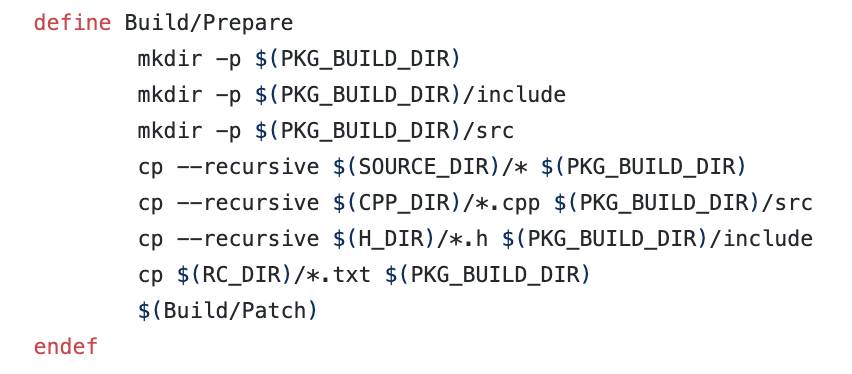


This is a continuation of the package description. You can add much more writing here to describe the package.

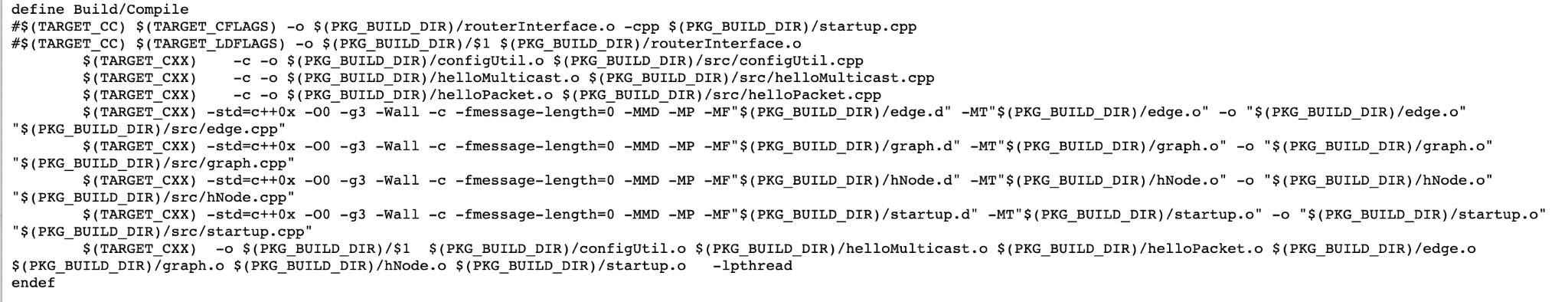


This part of the makefile defines what needs to happen to prepare for the package to be built. PKG\_BUILD\_DIR is where the build system builds the package from. We do not define this, if the build system was set up correctly it should be in the file “build\_dir”

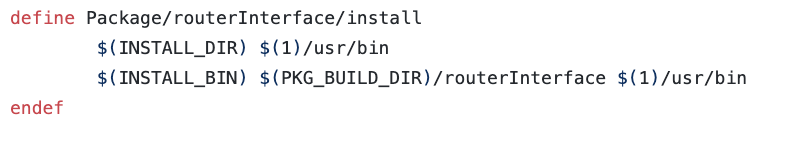
These commands below create the build directory and copy the necessary files from the project into it. (the executable files, .cpp files, .h files, and router configuration file)



The section of the makefile is basically the makefile for the original source project except that it uses the target-specific compiler defined by the toolchain to compile the source code files, then link it to the final cross-compiled executable. It also instead of using the source project directories, uses the directories in the build directory specified by PKG\_BUILD\_DIR.



This creates a directory within the package to hold the executable and then copies the project executable we built into the folder.



# Build System Structure: Important Folders

**build\_dir:** used to unpack all the project source files and compile them.

**staging\_dir:** used to install the compiled program from build\_dir to be used to build packages or preparing the firmware image to be put on the router.

**mypackages/<SECTION>/<PKG\_NAME>**: this is where you should put the makefile for your package.

**source/bin/packages/x86\_x64/mypackages :** this is where your package (.ipk file) will be once it has been built. From here it is ready to copy over to your router.

**SEE FOR MORE INFORMATION**: <https://openwrt.org/docs/guide-developer/build-system/start>

# How to Cross Compile

Once you have added your package, you will need to run the commands from Part 3 of the hello world documentation (as shown below)

cd /home/deyo2303/buildroot/source

./scripts/feeds update mypackages

./scripts/feeds install -a -p mypackages

If this is successful, you should see a response in the terminal like “Installing package ‘helloworld’ from mypackages”. **If you DO NOT see this, check that the path you put in your feeds.conf and your package manifest files are correct.**

**You will need to follow Part 4 from the hello world documentation**

From the source folder, run:

make menuconfig

From the GUI that pops up:

1. scroll down to the examples menu and press enter
2. highlight the package called “<PKG\_NAME>” and press y. You should see a star next to it now
3. use the right arrow key until you are over “save”
4. press enter and follow the menus to save your configuration (save to the default .config file)
5. use the left arrow key until you are over “exit”
6. exit until you close the GUI

Now, run the following command:

make package/<PKG\_NAME>/compile

If you are successful you should see a package called **<PKG\_NAME>\_<VERSION>-<RELEASE>\_<target>.ipk** in your **/source/bin/packages/<target>/mypackages** folder.

You can now scp your package to the root of your router. You will also need to scp the router configuration file to the tmp directory of your router, unless of course you changed the location in the source code.

# Tips and Tricks

* Make sure to make clean before every build or you could use up all of your memory on your machine! This is especially important if you are using a virtual machine, which tends to have less partition space.

To do this you will need to run the command make clean from the buildroot directory

* When you run make package/<PKG\_NAME>/compile add V=sc in order to debug any failed builds. This will tell you why the build failed and whether or not you are missing any libraries.
* You will not be able to debug from the router. Whenever you make changes to your source code you will need to go through this process again. If you get a seg fault, make sure you have the configuration file in the right place.

If this does not solve your problem, install **strace** onto your router using opkg. This will let you see which system call resulted in the segmentation fault, even without using gdb. You would then enter the command

strace <PKG-NAME> <any arguments>

* For any further troubleshooting, refer to the OpenWrt documentation, or OpenWrt forums.