

# Eclipse Configuration for Nordic

This page is intended to give an introduction on how to set up Eclipse with the tools needed to develop code for [nrf51](#) and [nrf52](#) kits from [Nordic](#)

## Nomenclature

CDT c/c++ Development tools JDK Java Development Kit JRE Java Runtime Environment

## Configuring Eclipse with CDT

### Eclipse neon on windows 10

1. Download eclipse
  - a. start eclipse if it returns eclipse returned exit code 1. You need to point eclipse to the correct java version
  - b. Download the correct version of java jdk for eclipse (1.8 minimum for eclipse neon)
  - c. Open eclipse.ini located in the eclipse directory
  - d. Add the following above two lines before -vmargs: -vm and <Path to JDK>/bin
  - e. now eclipse should start
2. Install the GNU ARM Eclipse plug-in [\[1\]](#)
  - a. Open eclipse marketplace from help-> marketplace and drag the install from the link in.
  - b. Confirm and install
    - i. Download this from oracle.
    - ii. Extract and copy files into <path to JDK>/jre/lib/security
    - iii. Restart eclipse and try again
3. Check the global tools paths
  - a. Go to window->preferences->c/c++ ->build->global tools path and check if this is correct. (GNU Tools for ARM Embedded Processors)
4. Set workspace preferences [\[4\]](#). At least set the space formatting in the code formatter! (Remember that this will make you unable to use tab in makefiles.)
5. You should now be ready to go!

### Eclipse on Linux <xx>

Linux gurus write something here!

## Start a new Project

1. Find a suitable location for your project
2. Copy a config folder from one of the examples in the SDK to your project folder
3. Update your makefile to reflect these changes. This includes the PROJECT\_NAME, SDK\_ROOT, PROJ\_DIR, LINKER\_SCRIPT also some backslash needs to be removed or added depending on your chosen location.
4. Try making the project on the command line before trying to compile through Eclipse.
5. Import the project in Eclipse by file->new(alt+shift+n).
  - a. Name your project
  - b. Existing code location should be the location of your makefile for the project.
  - c. Select "Cross ARM GCC" from the list of toolchains.
  - d. Click finish and you are done.
6. Remember to set the build command to "make VERBOSE=1" this field can be found by marking the project, pressing alt+enter and navigate to c/c++ build -> build settings.
7. Try compiling.
8. Change the compiler command pattern under project properties-> c/c++ General-> Preprocessor Include Paths, Macros... -> providers -> CDT GCC Build output parser to "(.\*gcc)(\[.\*\])" This is for symbols to be handled correctly in Eclipse.
9. Try rebuilding the project and you should be good.
10. If symbols are not detected by the OCD add symbol by hand in properties->c/c++ General->Paths and Symbols -> symbols-> add. Do you find another solution please write it down here!

## Tips And Tricks for error messages

If you have forgotten to include paths to c files in the make file you would get undefined reference errors similar to: `main.c:139: undefined reference to `nrf_drv_timer_init'`

If you get error messages similar to this: `error: invalid suffix "_INSTANCE_INDEX" on integer constant` You probably defined something you shouldn't have.