

Eclipse CDT setup guide for darktable in Windows

I have been requested to post a guide about setting Eclipse CDT IDE for darktable, so here it is. This guide assumes that you are already able to build darktable from sources in Windows (through MSYS2), by following the build guide on GitHub.

Basic knowledge of Eclipse CDT is also required.

If you want to contribute to darktable project, normally you need to create your own fork of the project in GitHub.

We assume that darktable sources are in C:\msys64\home\<username>\develop\darktable

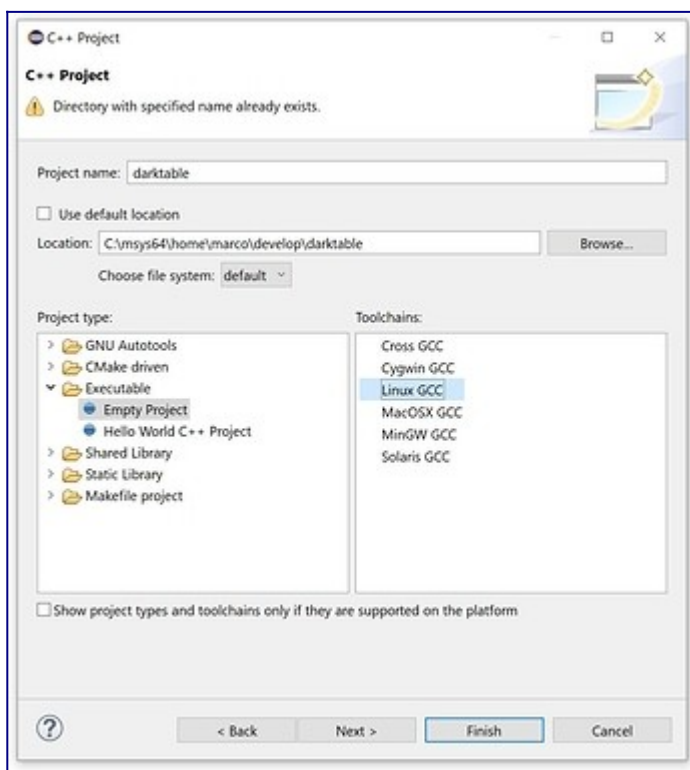
Step 1. Install Eclipse CDT from website

Step 2. From the Eclipse Marketplace, search and install the addon cmake4eclipse

Step 3. Import Darktable source code

In eclipse: File → New → Project → C/C++ → C++ Project

In the dialog, choose “Linux GCC toolchain” and point to the source code folder. See below

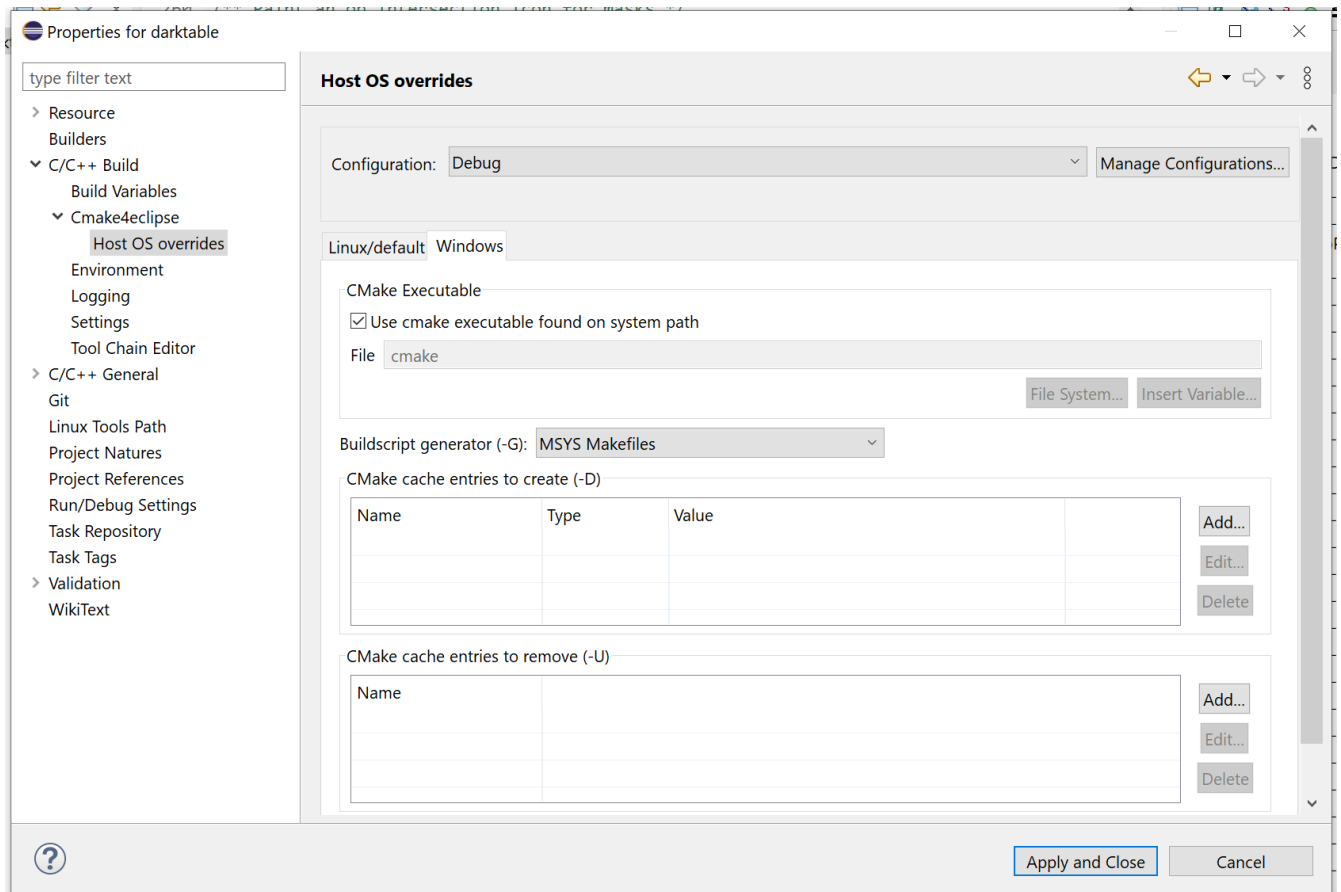
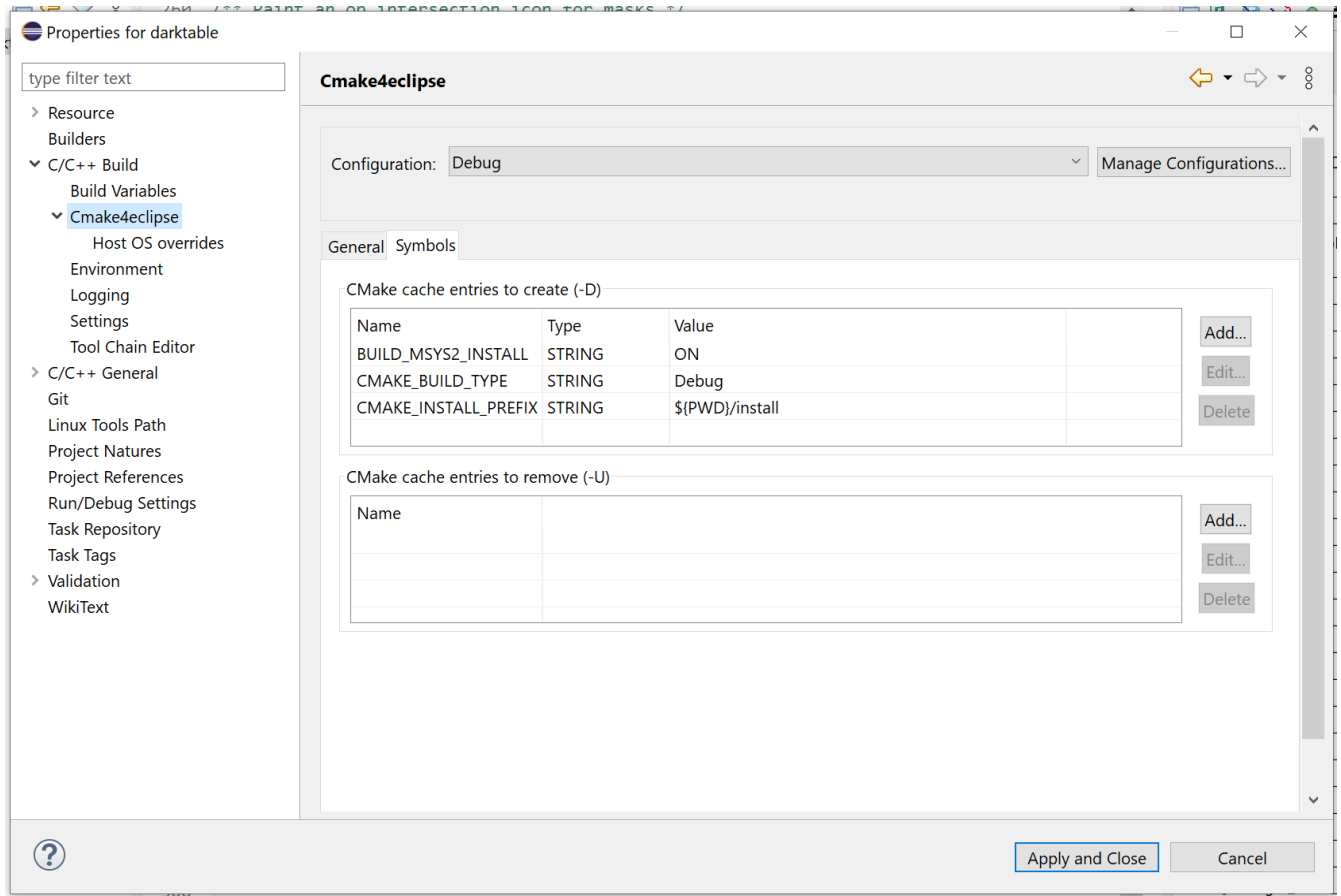


Hit “Finish”

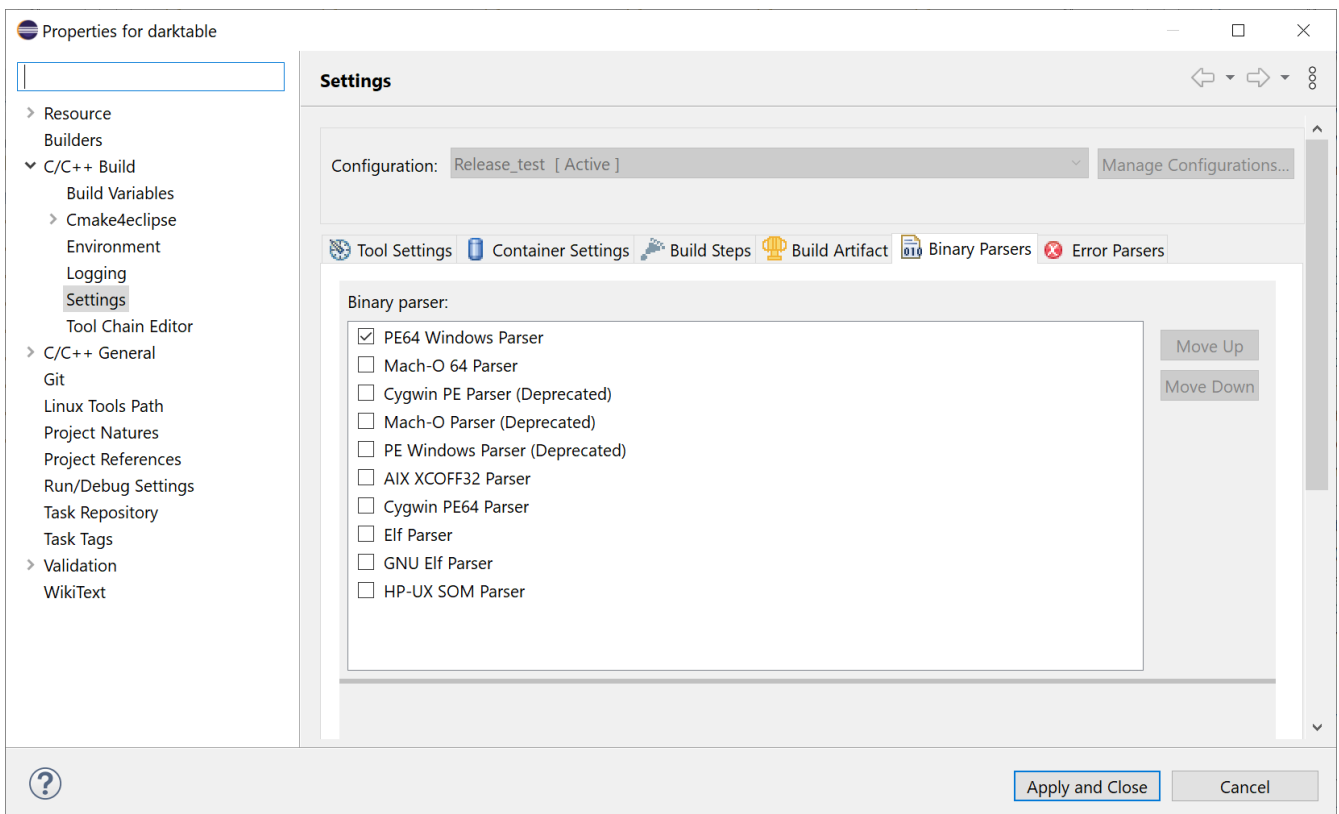
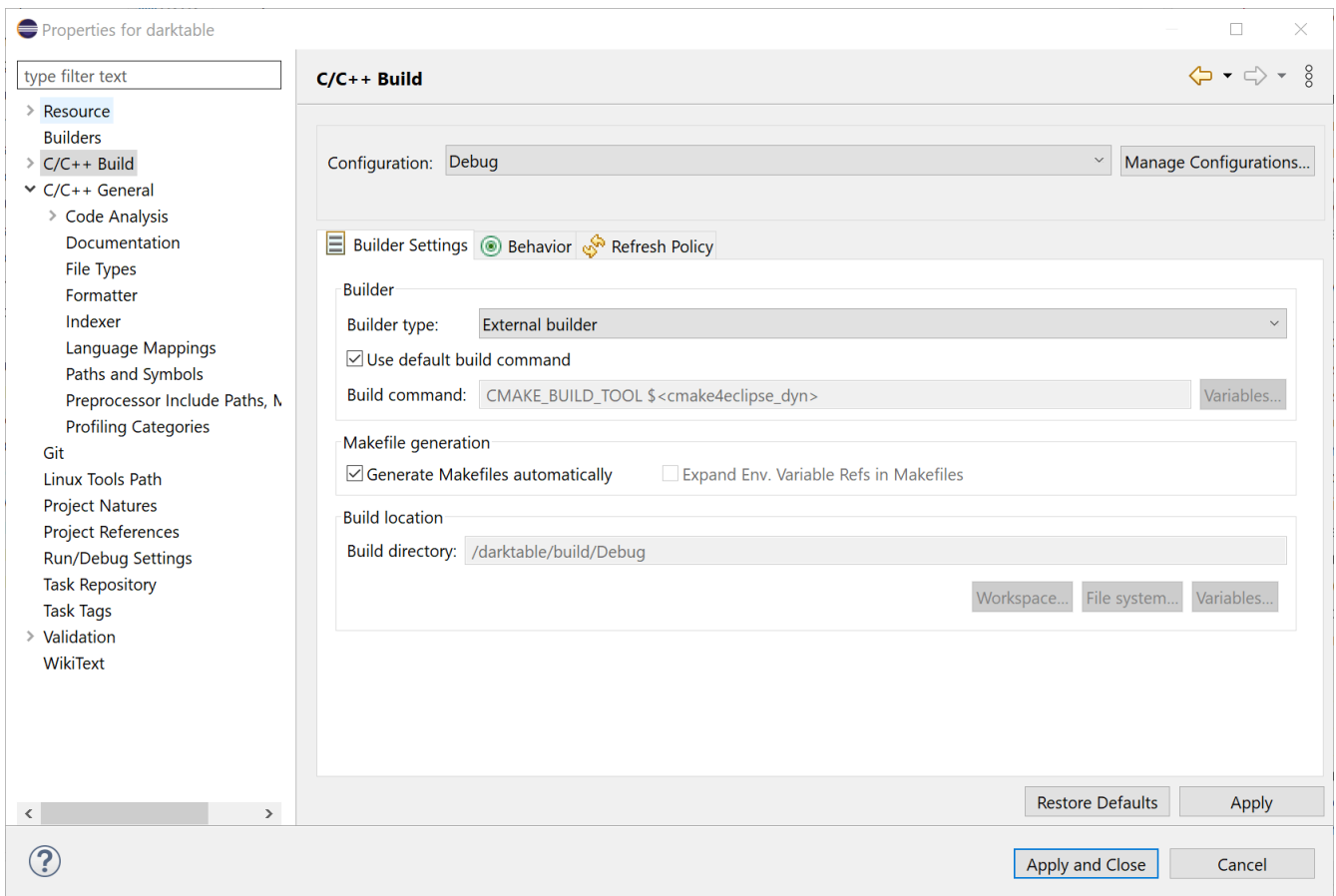
Step 4. Set Project Properties

Go to the properties of the newly created Eclipse project. For example in Project explorer window, right click on the project → Properties

Setup as follows (hit “Apply” at every step)



For newer releases of darktable you can also use Ninja as Buildscript generator, which is faster, provided you installed in MSYS2 the relative toolchain.



Hit “Apply and Close”

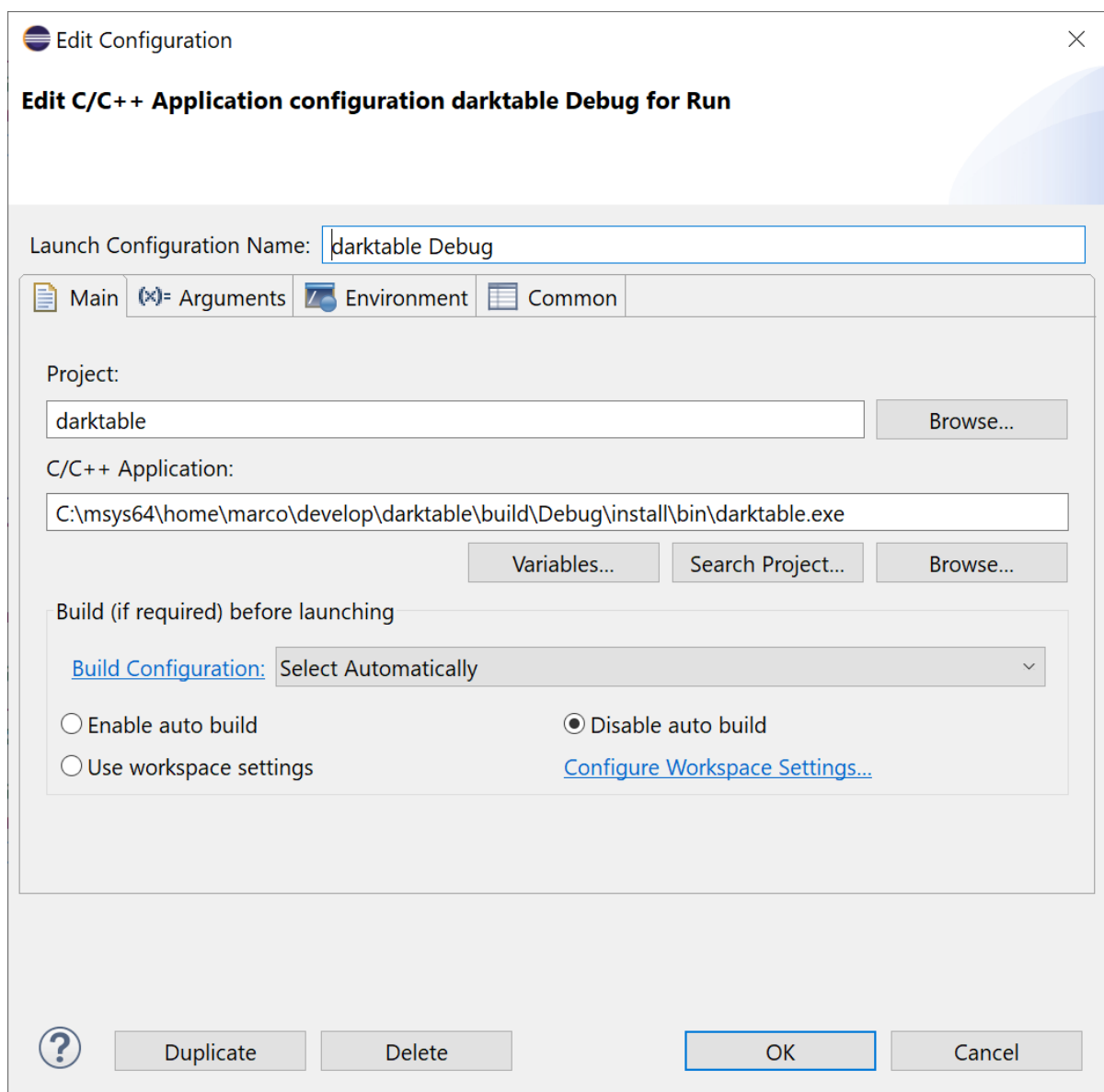
Step 5. Build the project

Eclipse should run CMake first and then make (or ninja) and install, you can see the output in the Eclipse console.

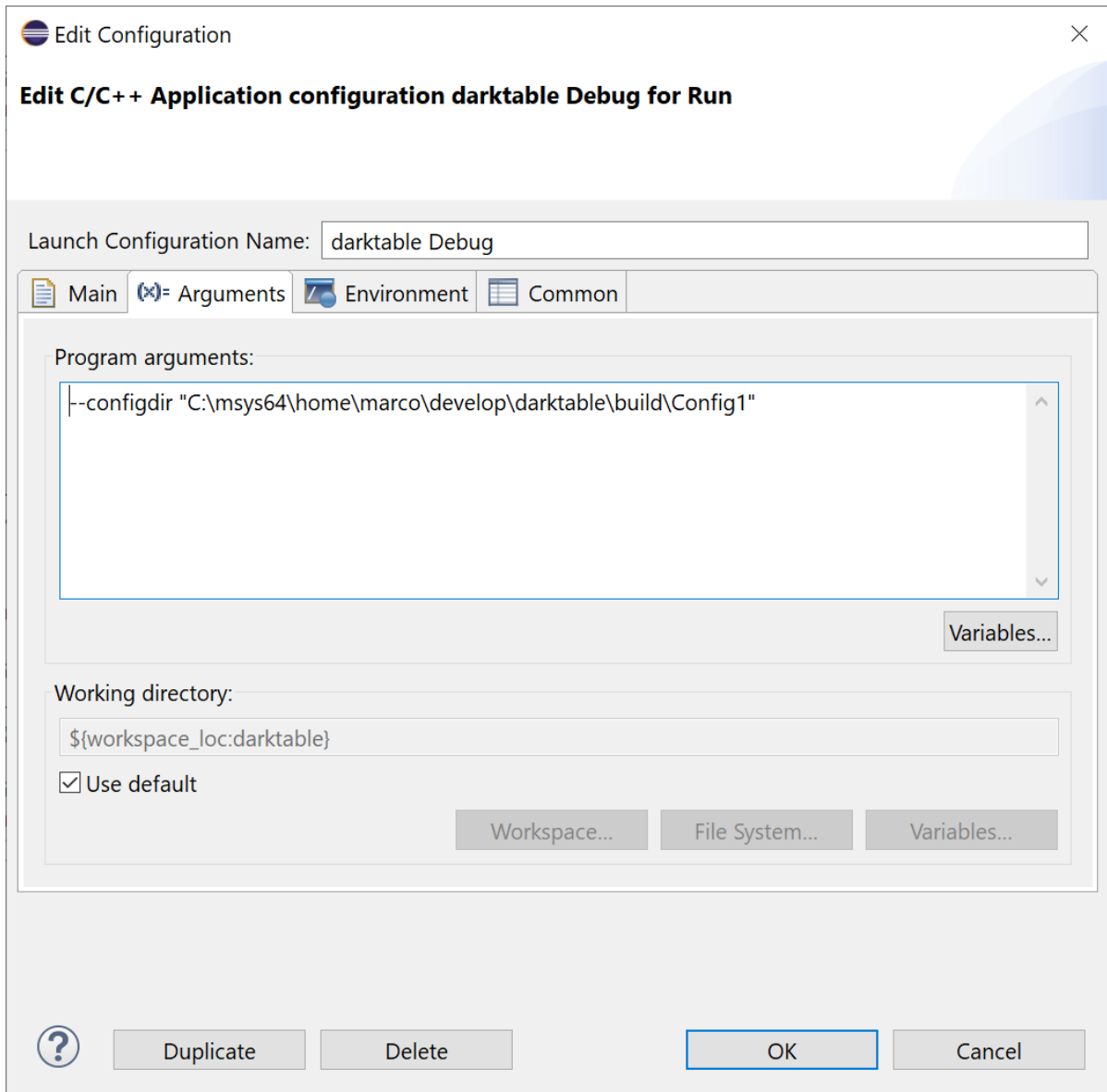
The initial build could give some errors related to failing build of user manual and/or translations, that should not be a problem.

Incremental builds shall not give errors.

Step 6. Create a run configuration

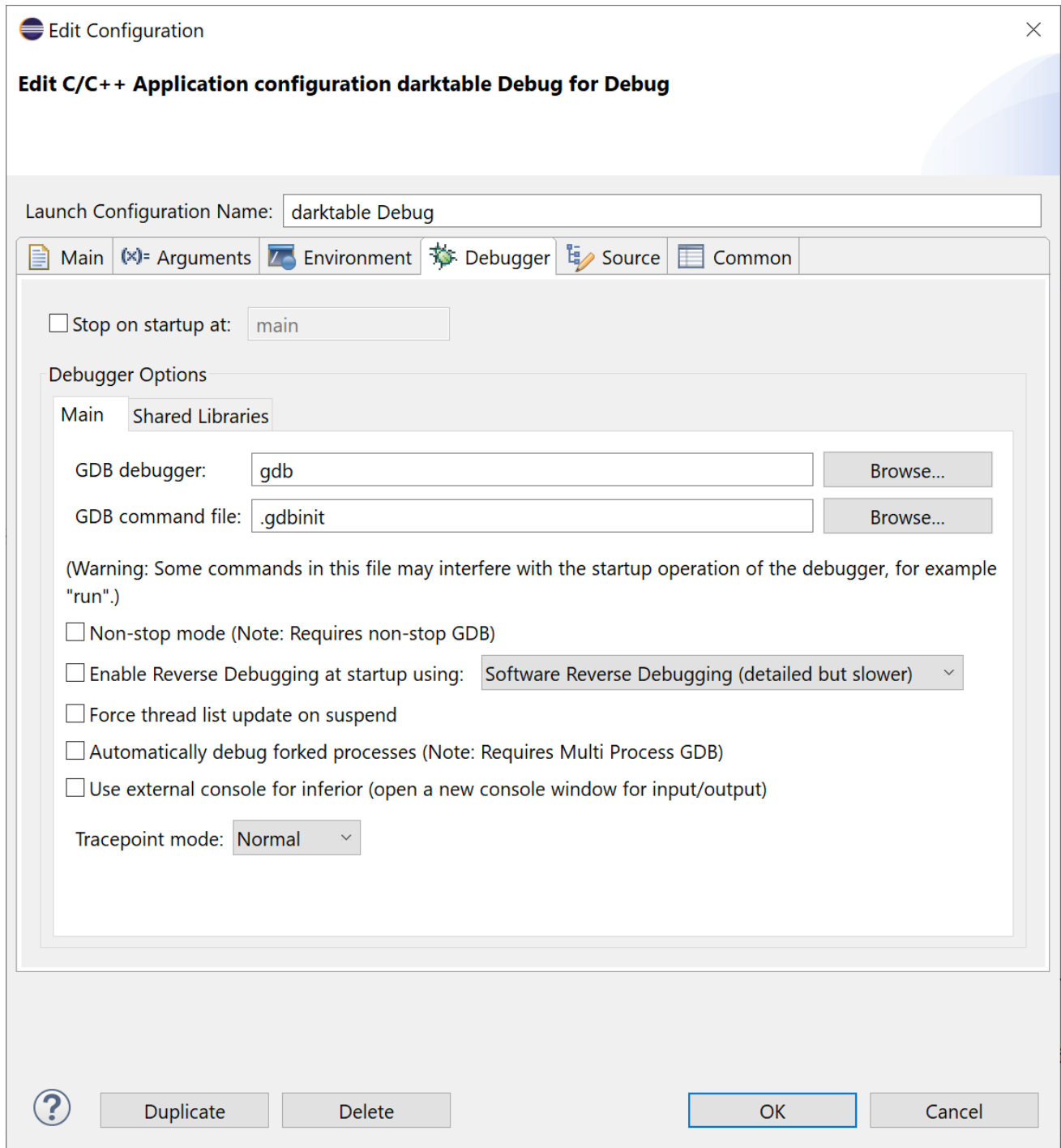


It is recommended you specify a configuration folder (you can have more than one)



In this dialog you can also specify additional command line arguments for darktable, like debugging options (-d <debug option>).

Step 7. Create a debug configuration



Now Eclipse should be able to run and debug darktable.

Final remarks

- You can define more configurations, like Release for example by changing `CMAKE_BUILD_TYPE` or other parameters.
- In general, building and debugging in Windows is slower than in Linux.

Feel free to comment / modify / improve.