# Creating OSCAR development environment on Windows and building installers

This document is intended to be brief notes on how to install the necessary components and create installers for Windows 32-bit and 64-bit versions of OSCAR. This documentation will be fleshed out later and probably placed in the Wiki for the OSCAR repository.

On my computers, I have QT installed in E:\QT and the OSCAR code base in E:\oscar\oscar-code. On another computer, they are on the F: drive. All references in the deploy.bat file are relative, so it should run with Oscar-code installed at any location.

**Required Programs**

The following programs and files are required to create Windows installers:

* Inno Setup 5.6.1 from <http://www.jrsoftware.org/isdl.php>. Download and install innosetup-qsp-5.6.1-unicode.exe.
* GIT for windows, from <https://gitforwindows.org/>. GIT for Windows adds itself to your path.
* Gawk is required. You can use the version included with Git for Windows or install Gawk for Windows from <http://gnuwin32.sourceforge.net/packages/gawk.htm>. The deployment batch file will use the Git for Windows version if gawk.exe is not in your PATH.
* QT Open Source edition from <https://www.qt.io/download>. I use version 5.12.2. More recent versions may also work but I have not tested any.

**Installing Inno Setup 5**

Inno Setup 5.6.1 is found on <http://www.jrsoftware.org/isdl.php>. Download and install innosetup-qsp-5.6.1-unicode.exe. Do not download 6.x versions. They are still in beta as this is written, and changes to the installer script may be required.

The deployment batch file assumes that Inno Setup is installed into its default location: C:\Program Files (x86)\Inno Setup 5. If you put it somewhere else, you will have to change the batch file.

Run the installer, accepting options to install inno script studio (for possible future use) and install Inno Setup Preprocessor. Encryption support is not needed, so do not select it.

**Installing GIT for Windows**

Go to <https://gitforwindows.org/> and click on the Download button. Run the installer, which presents lots of options:

* Select whichever editor you desire. for Windows adds itself to your path.
* Select “Use Git and optional Unix tools from the Command Prompt.” If you do this, rather than “Git from the command line and also from 3rd-party software,” you will not need to install Gawk separately, as it is included with Git for Windows.
* Select “Use the OpenSSL library.”
* Select “Checkout Windows-style, commit Unix-style line endings.”
* Select “Use Windows’ default console window.” I find the default console window satisfactory with Windows 10.
* Leave extra options as they default (enable file system caching, enable Git credential manager, but not symbolic links).

GIT for Windows adds itself to your path.

**Installing Gawk (if Git for Windows gawk is not used)**

From <http://gnuwin32.sourceforge.net/packages/gawk.htm>, download setup for “Complete package, except sources”. When downloaded, run the setup program. Accept default options and location. The deployment batch file assumes that gawk.exe is in your PATH, so either add c:\Program Files (x86)\gnuwin32\bin to your PATH or copy the executables to some other directory already in your PATH.

**Installing QT**

Go to QT at <https://www.qt.io/download> and download the Open Source edition of the Windows online installer, qt-unified-windows-x86-3.0.6-online.exe. Run the installer:

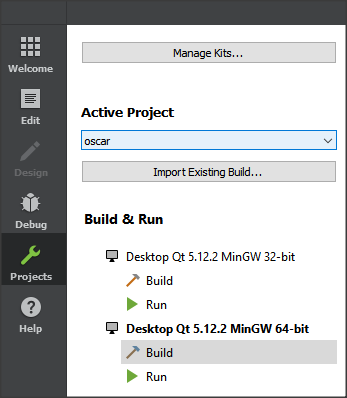
* Logon with your QT account or create an account if needed.
* Click Next to download meta information (this takes a while).
* Choose your installation directory (I picked F:\Qt, but there are no dependencies on where QT is located)
* Select components:
  + In QT 5.12.2:
    - MinGW 7.3.0 32-bit
    - MinGW 7.3.0 64-bit
  + In Developer and Designer Tools:
    - QT Creator 4.8.2 CDB Debug (this may not be required)
    - MinGW 7.3.0 32-bit
    - MinGW 7.3.0 64-bit

And complete the installation (this also takes a while).

**Getting Started**

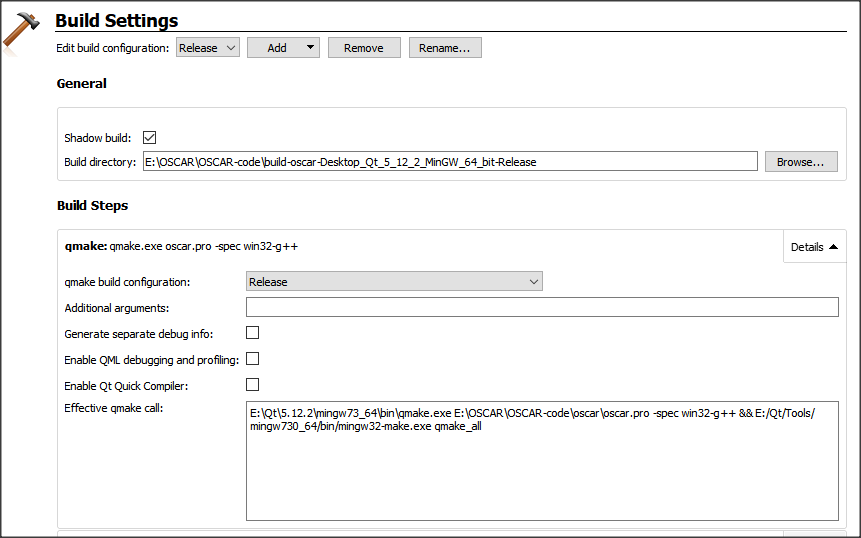
In browser, log into gitlab.com. (If you do not have a gitlab account and access to the OSCAR project, contact …).

When you start QT, it will ask you to select your kits and configure them. Select both MinGW 7.3.0 32-bit and 64-bit kits.

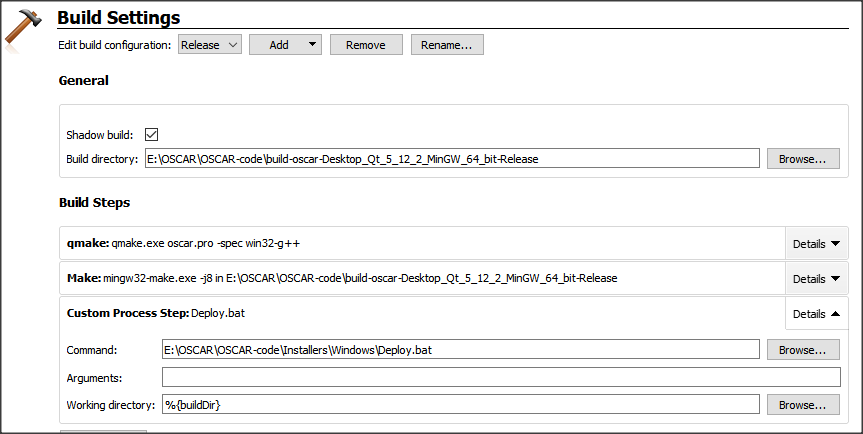


**Adjusting the Build Steps**

Select “Release” rather than the default “Debug” in the pull-down at the top of the Build Settings. Click on Details for the qmake build step. By default, “Enable Qt Quick Compiler” is checked. Remove that check – errors result if it is on. QT will ask if you want to recompile everything now. Don’t there is more to do. Make this same change for the Release build for both 32-bit and 64-bit kits. And if you ever want to use the Build Debug pull-down, you probably need to do the same (I don’t remember and haven’t used Debug in a while).



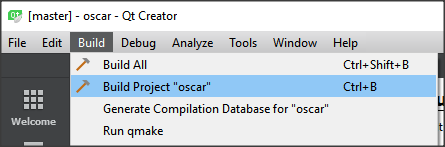
To create a “release” directory and installer, you need to add a custom process step to the Release build configuration. Be sure to select “Release” rather than the default “Debug” in the pull-down at the top of the Build Settings:



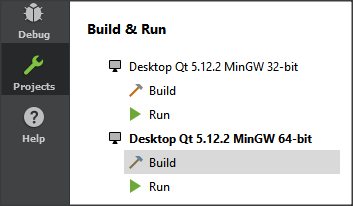
Create a custom process step as the final build step. Put the fully qualified path for deploy.bat in the command field. Don’t touch “working directory.” Any string you care to place in the “arguments” field will be appended to the installer executable file name.

Do the same for both 32-bit and 64-bit Build settings.

Now you should be able to build OSCAR:



To make 32-bit or 64-bit builds, just make sure the correct Build item is selected in the Build & Run section on the left:



**The Deploy.BAT file**

The deployment batch file creates two folders inside the shadow build folder:

Release – everything needed to run OSCAR. You can run OSCAR from this directory just by clicking on it.

Installer – contains an installer exe file that will install this product.