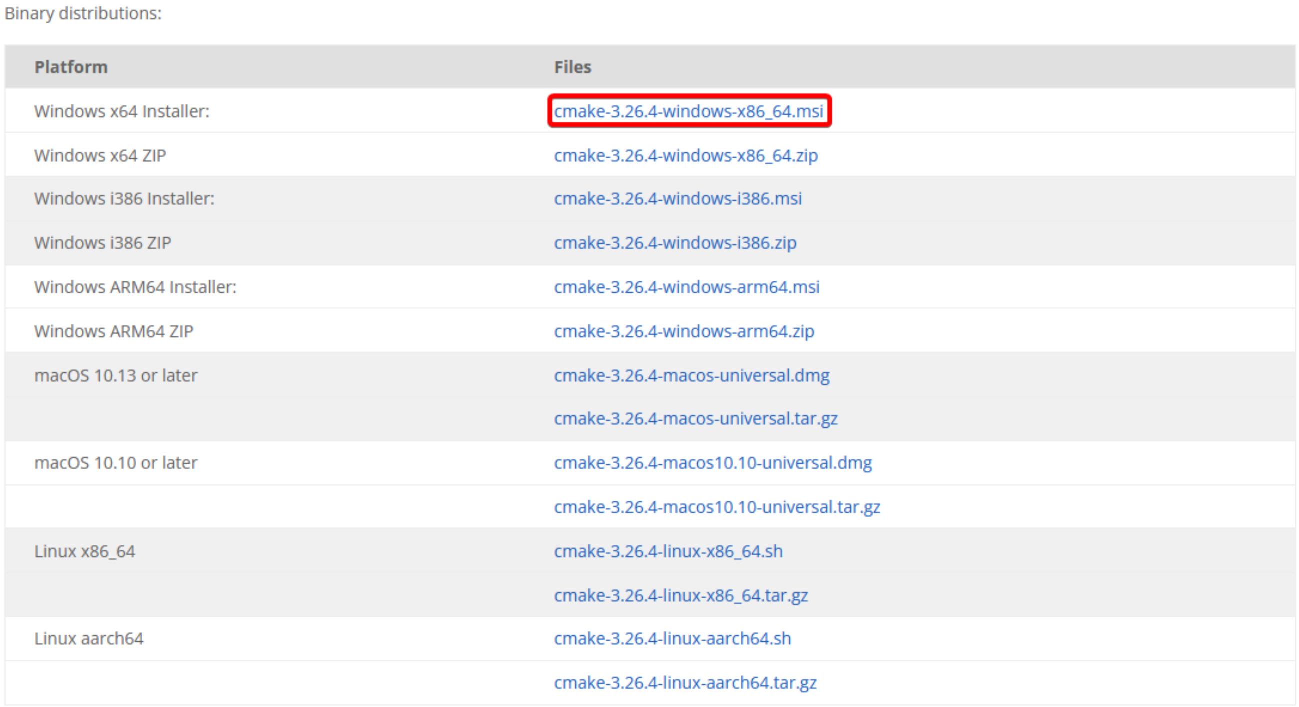
Building for Windows

MSVC Build for Windows

Minimal Dependencies

On Windows, all library dependencies are automatically included within the “externals” folder or can be downloaded on-demand. To build Citra, you simply need to install:

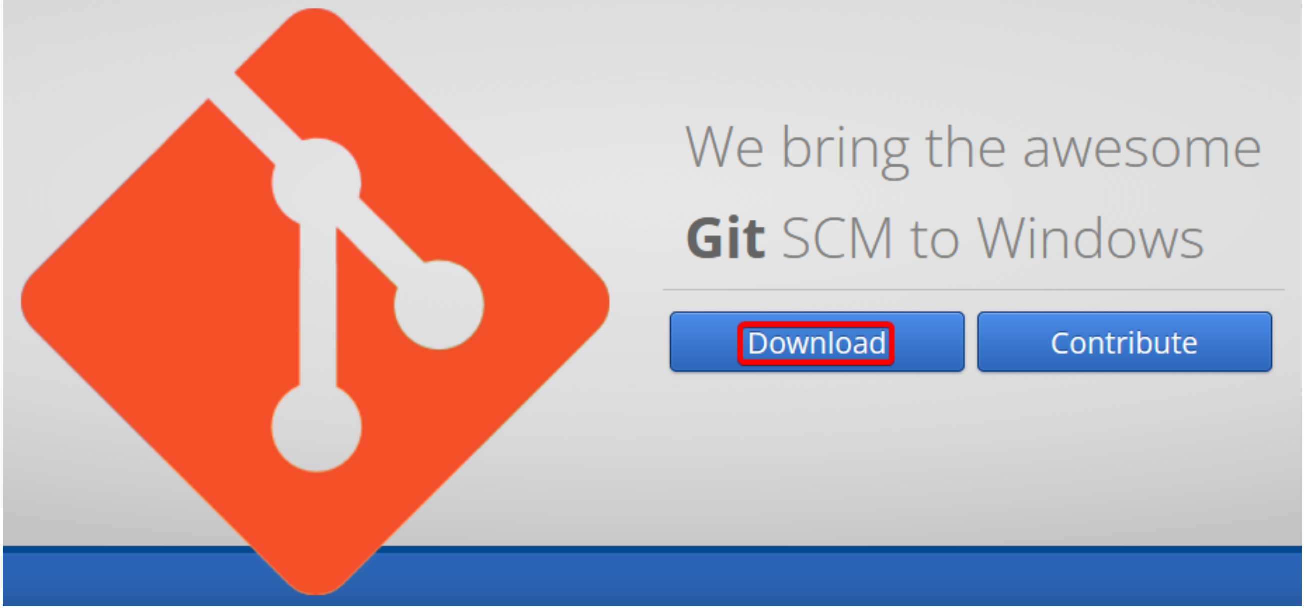
- **Visual Studio 2022 (https://visualstudio.microsoft.com/vs/)** - **Make sure to select C++ support in the installer**.
- **CMake (https://cmake.org/download/)** - Used to generate Visual Studio project files.



Binary distributions:

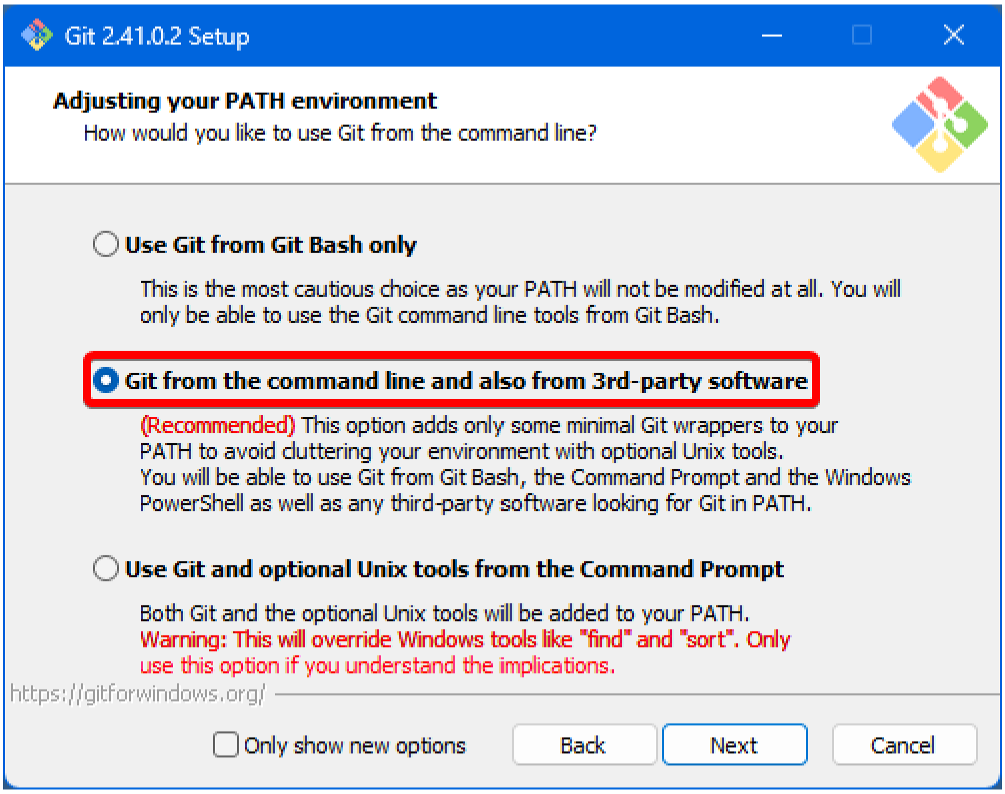
Platform	Files
Windows x64 Installer:	cmake-3.26.4-windows-x86_64.msi
Windows x64 ZIP	cmake-3.26.4-windows-x86_64.zip
Windows i386 Installer:	cmake-3.26.4-windows-i386.msi
Windows i386 ZIP	cmake-3.26.4-windows-i386.zip
Windows ARM64 Installer:	cmake-3.26.4-windows-arm64.msi
Windows ARM64 ZIP	cmake-3.26.4-windows-arm64.zip
macOS 10.13 or later	cmake-3.26.4-macos-universal.dmg cmake-3.26.4-macos-universal.tar.gz
macOS 10.10 or later	cmake-3.26.4-macos10.10-universal.dmg
	cmake-3.26.4-macos10.10-universal.tar.gz
Linux x86_64	cmake-3.26.4-linux-x86_64.sh cmake-3.26.4-linux-x86_64.tar.gz
Linux aarch64	cmake-3.26.4-linux-aarch64.sh
	cmake-3.26.4-linux-aarch64.tar.gz

- **Git** - We recommend Git for Windows (https://gitforwindows.org/).


We bring the awesome

Git SCM to Windows

- While installing Git Bash, you should tell it to include Git in your system path. (Choose the “Git from the command line and also from 3rd-party software” option.) If you missed that, don’t worry, you’ll just have to manually tell CMake where your git.exe is, since it’s used to include version info into the built executable.


Git 2.41.0.2 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

☐ Only show new options

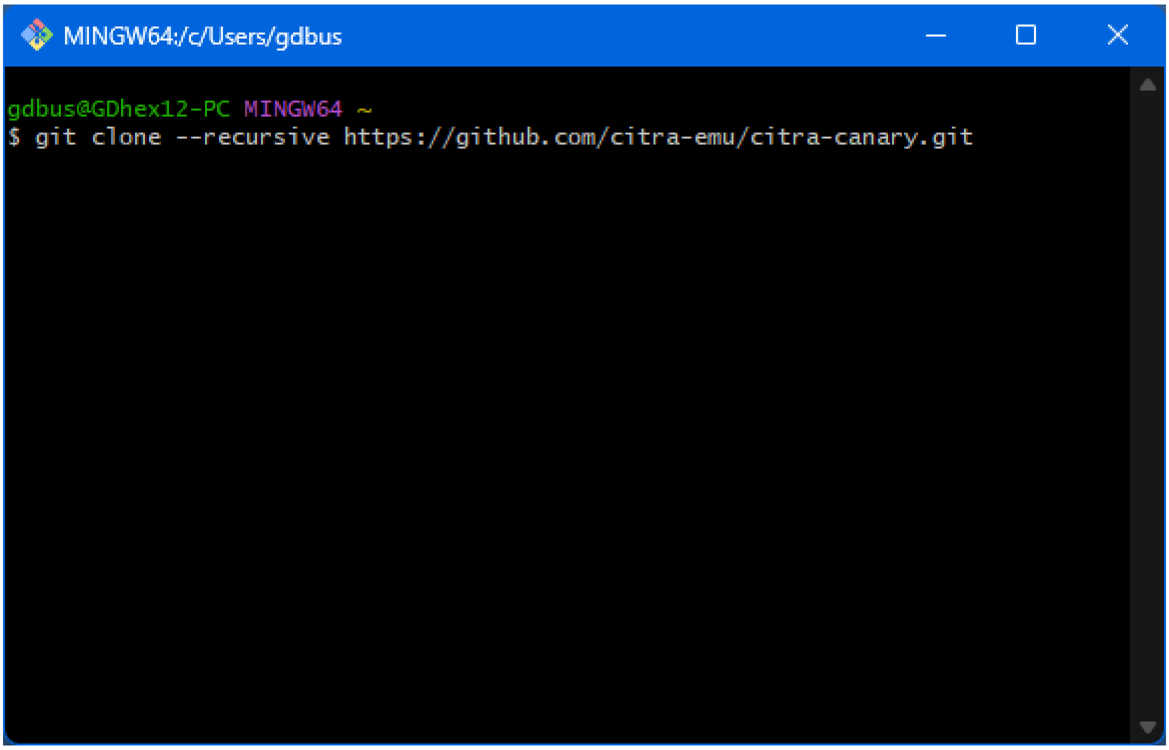
Cloning Citra in Git:

Master:

```
git clone --recursive https://github.com/citra-emu/citra.git
cd citra
```

Canary:

```
git clone --recursive https://github.com/citra-emu/citra-canary.git
cd citra-canary
```

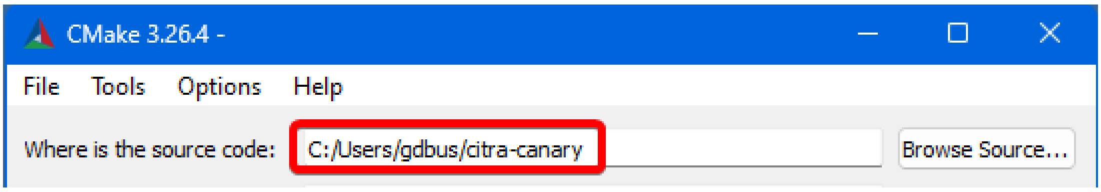


```
gdbus@GDhex12-PC MINGW64 ~
$ git clone --recursive https://github.com/citra-emu/citra-canary.git
```

(Note: Citra by default downloads to `C:\Users\\citra` *(Master) or* `C:\Users\\citra-canary` *(Canary)*

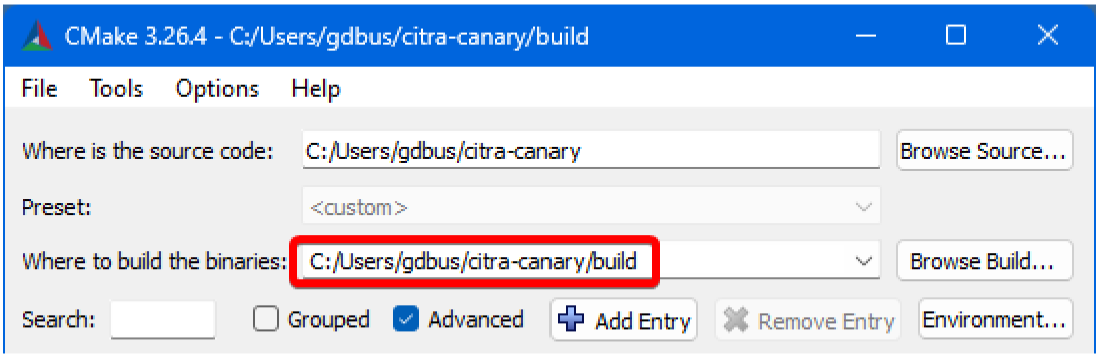
Building

- Open the CMake GUI application and point it to the `citra` (Master) or `citra-canary` (Canary) directory.



Where is the source code: C:/Users/gdbus/citra-canary

- For the build directory, use a `build/` subdirectory inside the source directory or some other directory of your choice. (Tell CMake to create it.)



Where is the source code: C:/Users/gdbus/citra-canary

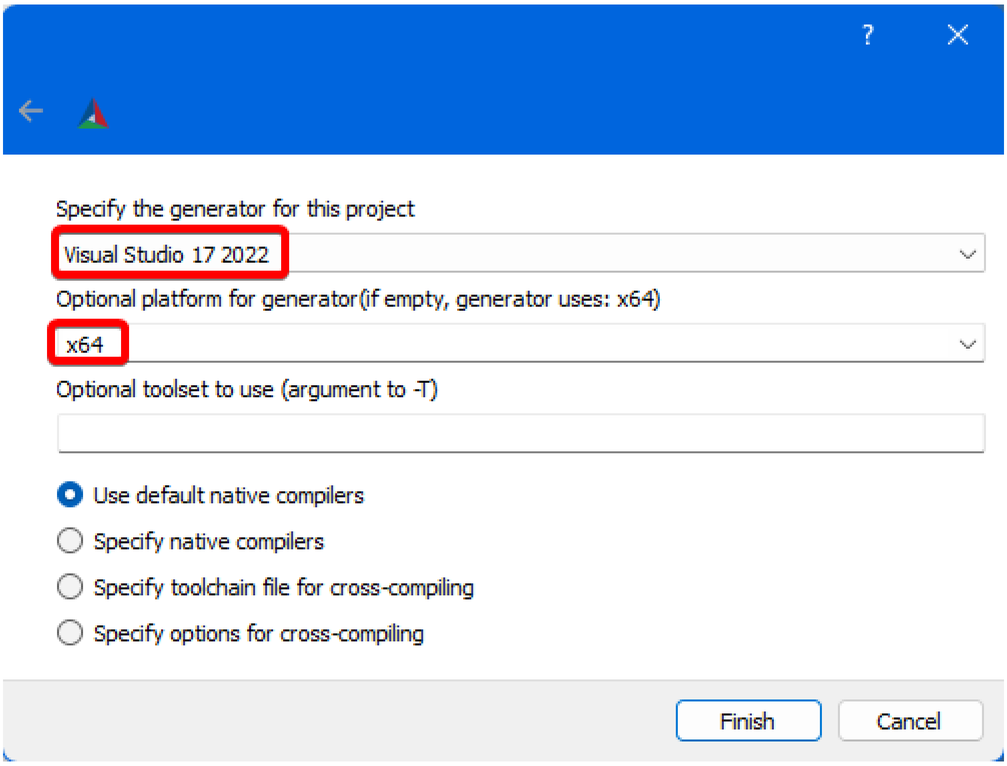
Preset: <custom>

Where to build the binaries: C:/Users/gdbus/citra-canary/build

Search:

☐ Grouped ☑ Advanced

- Click the "Configure" button and choose `Visual Studio 17 2022`, with `x64` for the optional platform.


Specify the generator for this project

Visual Studio 17 2022

Optional platform for generator(if empty, generator uses: x64)

x64

Optional toolset to use (argument to -T)

● Use default native compilers

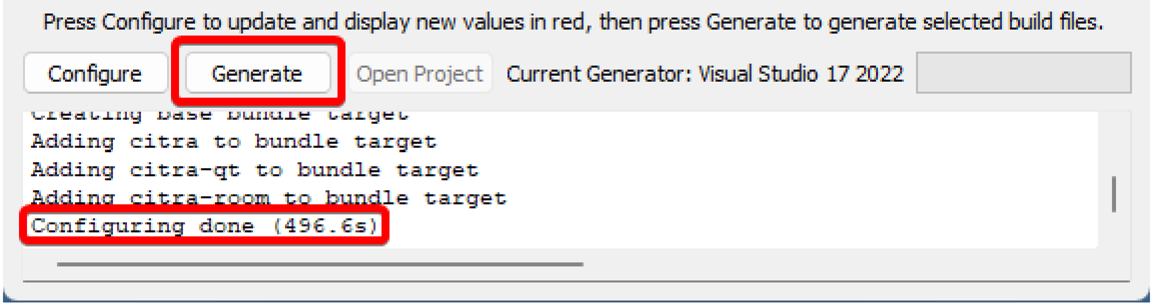
○ Specify native compilers

○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

```
* NOTE: If you get errors like "XXX does not contain a CMakeLists.txt file" at this s
tep, it means you didn't use the `--recursive` flag in the clone step, or you used to
ols other than the git CLI. Please run "git submodule update --init --recursive" to g
et remaining dependencies.
```

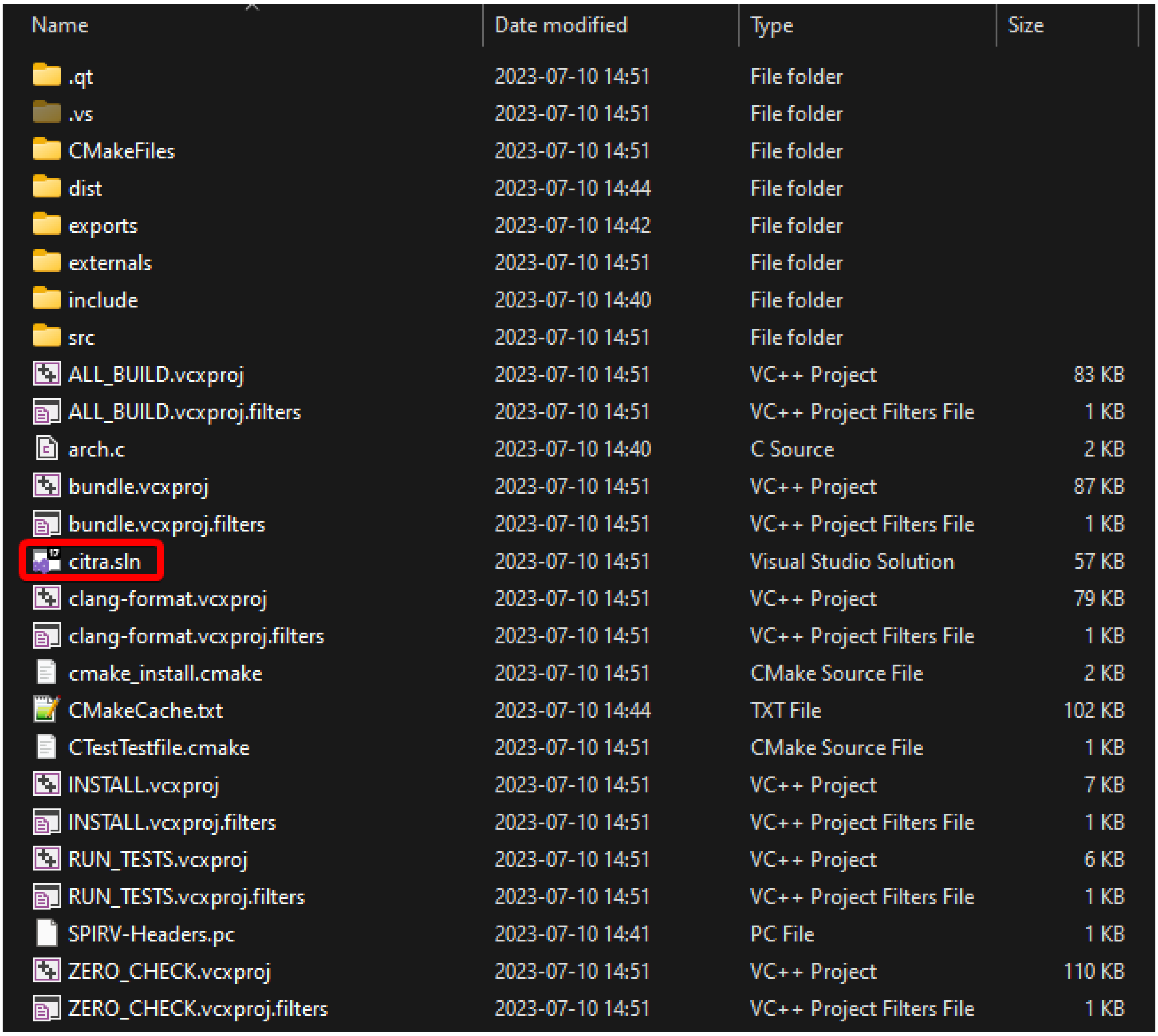
- Click “Generate” to create the project files.


Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 17 2022

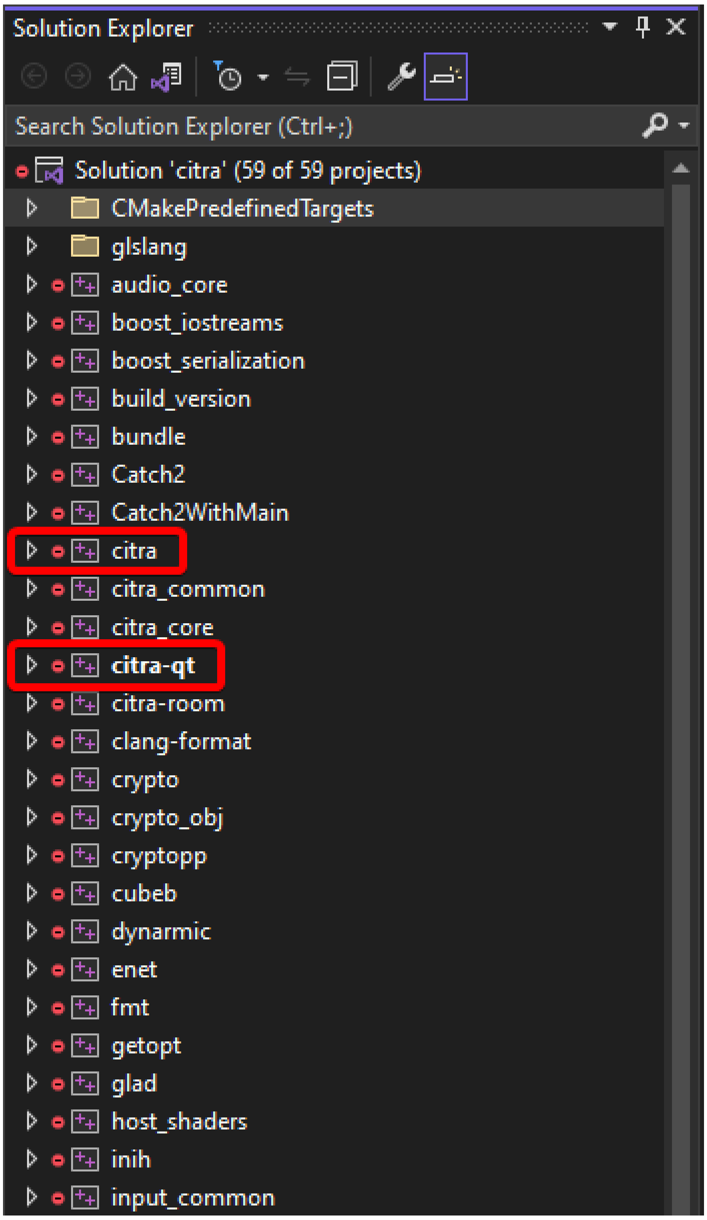
```
Creating base bundle target
Adding citra to bundle target
Adding citra-qt to bundle target
Adding citra-room to bundle target
Configuring done (496.6s)
```

- Open the solution file citra.sln in Visual Studio 2022, which is located in the build folder.

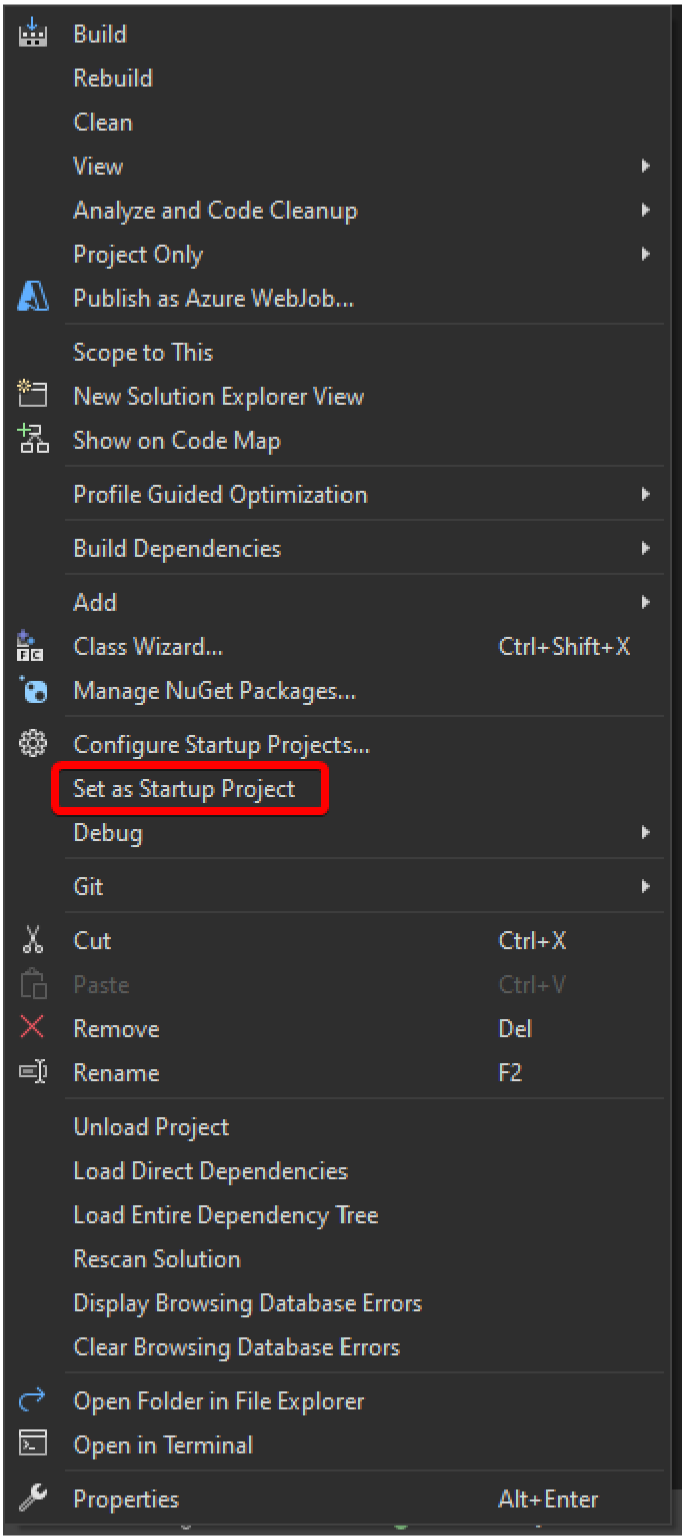


Name	Date modified	Type	Size
.qt	2023-07-10 14:51	File folder	
.vs	2023-07-10 14:51	File folder	
CMakeFiles	2023-07-10 14:51	File folder	
dist	2023-07-10 14:44	File folder	
exports	2023-07-10 14:42	File folder	
externals	2023-07-10 14:51	File folder	
include	2023-07-10 14:40	File folder	
src	2023-07-10 14:51	File folder	
ALL_BUILD.vcxproj	2023-07-10 14:51	VC++ Project	83 KB
ALL_BUILD.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB
arch.c	2023-07-10 14:40	C Source	2 KB
bundle.vcxproj	2023-07-10 14:51	VC++ Project	87 KB
bundle.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB
citra.sln	2023-07-10 14:51	Visual Studio Solution	57 KB
clang-format.vcxproj	2023-07-10 14:51	VC++ Project	79 KB
clang-format.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB
cmake_install.cmake	2023-07-10 14:51	CMake Source File	2 KB
CMakeCache.txt	2023-07-10 14:44	TXT File	102 KB
CTestTestfile.cmake	2023-07-10 14:51	CMake Source File	1 KB
INSTALL.vcxproj	2023-07-10 14:51	VC++ Project	7 KB
INSTALL.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB
RUN_TESTS.vcxproj	2023-07-10 14:51	VC++ Project	6 KB
RUN_TESTS.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB
SPIRV-Headers.pc	2023-07-10 14:41	PC File	1 KB
ZERO_CHECK.vcxproj	2023-07-10 14:51	VC++ Project	110 KB
ZERO_CHECK.vcxproj.filters	2023-07-10 14:51	VC++ Project Filters File	1 KB

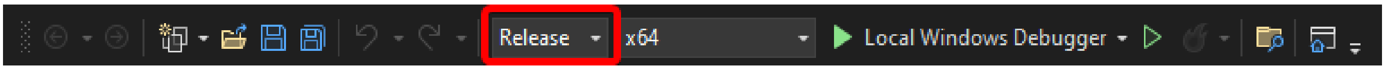
- Depending on which frontend (SDL2 or Qt) you want to build or run, select “citra” or “citra-qt” in the Solution Explorer, right-click and “Set as Startup Project”.



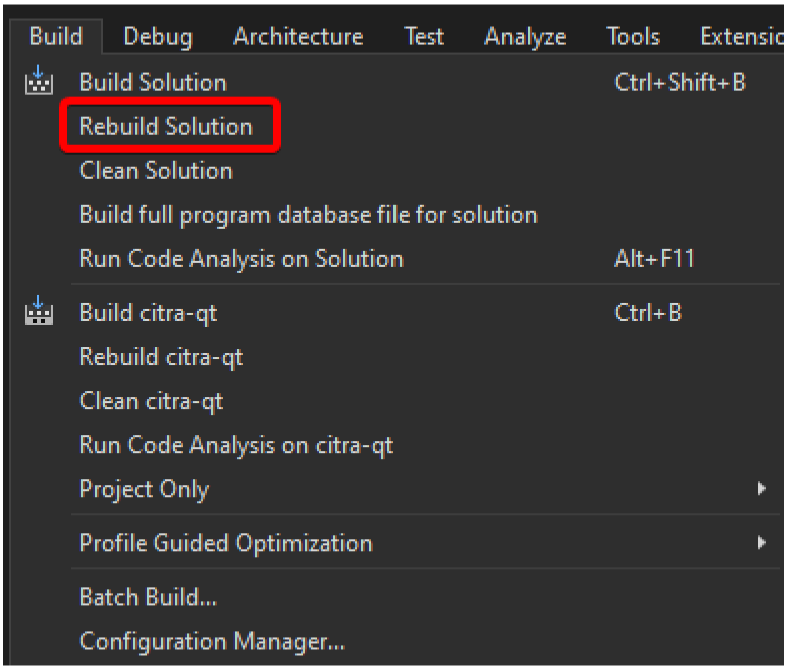
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'citra' (59 of 59 projects)
CMakePredefinedTargets
glslang
audio_core
boost_iostreams
boost_serialization
build_version
bundle
Catch2
Catch2WithMain
citra
citra_common
citra_core
citra-qt
citra-room
clang-format
crypto
crypto_obj
cryptopp
cubeb
dynarmic
enet
fmt
getopt
glad
host_shaders
inih
input_common



- Select the appropriate build type, Debug for debug purposes or Release for performance (in case of doubt choose the latter).



- Press F5 or select Build → Rebuild Solution in the menu.



- NOTE: Please refer to Common Issues (https://github.com/citra-emu/citra/wiki/Common-Issues) if any errors. If you did not find a solution feel free to ask us in the IRC channel: #citra @ Freenode (https://webchat.freenode.net/).

Clang Build with MSYS2

Prerequisites to install

- MSYS2 (http://msys2.github.io/)

Make sure to follow the instructions and update to the latest version by running `pacman -Syu` as many times as needed.

Install Citra dependencies

- Open the "MSYS2 Clang64" (clang64.exe) shell
- Download and install all dependencies using: `pacman -S mingw-w64-clang-x86_64-{gcc,qt6,cmake} make git`

Clone the Citra repository with git.

- `git clone --recursive https://github.com/citra-emu/citra.git`
- `cd citra`

Run the following commands to build Citra (dynamic linked build)

```
mkdir build && cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
make -j4
# test Citra out with
./bin/citra-qt.exe
```

Note! This build is not a static build meaning that you need to include all of the dlls with the exe in order to use it.

Building without Qt (Optional)

Doesn't require the rather large Qt dependency, but you will lack a GUI frontend.

- Pass the `-DENABLE_QT=no` flag to cmake

Citra

Blog (/entry)

Downloads (/download/)

Screenshots (/screenshots/)

Patreon (https://www.patreon.com/citraemu)

Donate (/donate/)

Documentation

Help Documents (/help/)

Compatibility (/game/)

Wiki (/wiki/home/)

FAQ (/wiki/faq/)

Blog

News & Articles (/entry)

RSS 2.0 (https://citra-emu.org/index.xml)

Social

YouTube (https://www.youtube.com/channel/UC_dcdgzuapBtAY4ol3x-90Q)

Facebook (https://www.facebook.com/citra.emu)

Twitter (https://twitter.com/citraemu)

Forums (https://community.citra-emu.org/)

Discord (/discord)

IRC (/chat)

Get Involved

GitHub / Source (https://github.com/citra-emu/citra)

Issues (https://github.com/citra-emu/citra/issues)

Pull Requests (https://github.com/citra-emu/citra/pulls)