



To get GNU Make on Windows, you can use one of these methods:

1. Use WSL (Windows Subsystem for Linux):

WSL allows you to run a Linux distribution alongside your Windows installation. This is the recommended way to get GNU Make and other Unix-based tools on your Windows machine, as it provides a full Linux environment.

To install WSL:

- a. Open PowerShell as Administrator and run this command to enable WSL:

css

 Copy code

```
wsl --install
```

- b. Restart your machine when prompted.

- c. After rebooting, open the Microsoft Store, search for your preferred Linux distribution (e.g., Ubuntu), and click Install.

- d. Launch the Linux distribution from the Start menu, and follow the on-screen prompts to set up your Linux user account.

- e. Update your Linux distribution's package list and install GNU Make:

go

 Copy code

```
sudo apt update sudo apt install make
```

2. Use MinGW or MSYS2:

MinGW (Minimalist GNU for Windows) and MSYS2 are software distributions that provide a minimal environment for building native Windows applications using the GNU toolchain. They both include GNU Make.

To install MinGW:

- a. Download the MinGW installer from <https://mingw-w64.org/> and run it.

- b. Follow the installation instructions, and make sure to include the "make" package.

- c. Add the MinGW binary directory to your Windows PATH environment variable:

Right-click on Computer, go to Properties > Advanced System Settings > Environment Variables.

Edit the "Path" variable and add the path to your MinGW binaries (e.g., **C:\MinGW\bin**).

To install MSYS2:

- a. Download the MSYS2 installer from <https://www.msys2.org/> and run it.
- b. Follow the installation instructions.
- c. Open the MSYS2 terminal, and install the "make" package using the following command:

```
go
```

 Copy code

```
pacman -S make
```

- d. Add the MSYS2 binary directory to your Windows PATH environment variable, as described in the MinGW installation instructions.

3. Use GnuWin32:

GnuWin32 provides native ports of popular GNU tools for Windows. To install GNU Make using GnuWin32:

- a. Download the Make installer for Windows from <http://gnuwin32.sourceforge.net/packages/make.htm>.
- b. Run the installer and follow the installation instructions.
- c. Add the GnuWin32 binary directory to your Windows PATH environment variable, as described in the MinGW installation instructions.

After installing GNU Make using one of these methods, you can use it from the Windows Command Prompt, PowerShell, or the integrated terminal in your favorite IDE.



why make cant find any makefiles when they was generated into folder by cmake ?



When you use CMake to generate Makefiles, they are typically placed in a build directory. If you run **make** from another directory, it will not find the Makefiles because it looks for them in the current working directory by default.

To fix this issue, you can either:

1. Change the working directory to the build directory before running **make**:

bash

 Copy code

```
cd path/to/build/directory make
```

2. Or, specify the path to the Makefile when running **make** by using the **-f** or **--file** option:

bash

 Copy code

```
make -f path/to/build/directory/Makefile
```

For example, if your project directory structure looks like this:

objectivec

 Copy code

```
my_project/ └─ CMakeLists.txt └─ build/ └─ Makefile
```

You should navigate to the **build** directory (**cd my_project/build**) before running **make**. Alternatively, you can run **make -f my_project/build/Makefile** from the parent directory (**my_project**).



umm, it seems that "Makefile" is missing even that cmake was run from build folder...



If the **Makefile** is missing even though you ran CMake from the build folder, it could be due to one or more of the following reasons:

1. CMake encountered an error during the generation process. To diagnose this issue, re-run CMake and carefully inspect the output for any error messages or warnings. Make sure you are using the correct command for CMake, which is:

css

 Copy code

```
cmake -S path/to/source/directory -B path/to/build/directory
```



After running this command, navigate to the build directory and check for the presence of the **Makefile**.

2. The generator specified in CMake is not set to produce Unix Makefiles. By default, CMake chooses the most suitable generator for your platform. On Windows, it might choose Visual Studio or another build system instead of Unix Makefiles. To force CMake to generate Unix Makefiles, run:

```
bash
```

 Copy code

```
cmake -G "Unix Makefiles" -S path/to/source/directory -B  
path/to/build/directory
```

This command will generate Makefiles in the build directory.

3. The CMakeLists.txt file in your source directory is not configured correctly, causing CMake to fail in generating the Makefile. Review your CMakeLists.txt file for any syntax errors, missing files, or incorrect paths. Ensure that it contains the appropriate commands for defining your project and targets.

If you still can't find the Makefile after checking these possibilities, please provide more information about your project, the contents of your CMakeLists.txt file, and any error messages or warnings that you encountered during the CMake process.
