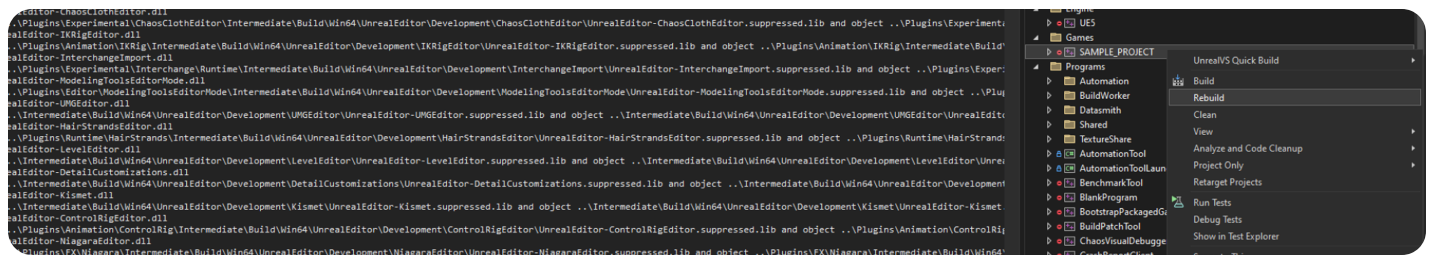


- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Programming and Scripting
- / Development Setup
- / Compiling Game Projects

Compiling Game Projects

Compiling individual game projects using Visual Studio on Windows, or Xcode on Mac.



Unreal Engine (UE) uses a custom building method using the **UnrealBuildTool (UBT)** which handles all the complex aspects of compiling the project and linking it with the engine. This process occurs transparently allowing you to simply build the project through the standard **Visual Studio** build workflow.

UnrealBuildTool uses the `*.build.cs` and `*.target.cs` files to build the game project. These are automatically generated when a project is created using a C++ template, or when the [CPP Class Wizard](#) is used to add code to a project created from a Blueprints Only template.

Build Configuration

Unreal projects have multiple targets, including **Editor**, **Client**, **Game**, and **Server**, described by `*.target.cs` files. Furthermore, each of these can be built to different configurations. In Visual Studio, this manifests as a Visual Studio `*.vcxproj` file with different configurations for each target. The solution configurations are named as **[Configuration][Target Type]** (for example, "DevelopmentEditor" for the default editor target, and "Development" for the default standalone game target). The configuration you use will be determined by the purposes of the build you want to create.

Every build configuration contains two keywords. The first keyword indicates the state of the engine and your game project. For instance, if you compile using a **Debug** configuration, the build process forgoes optimization making it easier to debug. To be clear, every configuration, even Shipping builds, produce symbols for debugging if built from Visual Studio or if **Project Settings > Project > Packaging > Project > Include Debug Files** is turned on in the Unreal Editor. This means that you can still debug Development and Shipping configurations, they just may not be as easy to debug as the Debug configuration. The second keyword indicates the target you are building for. For example, if you want to open a project in Unreal, you need to build with the **Editor** target keyword.

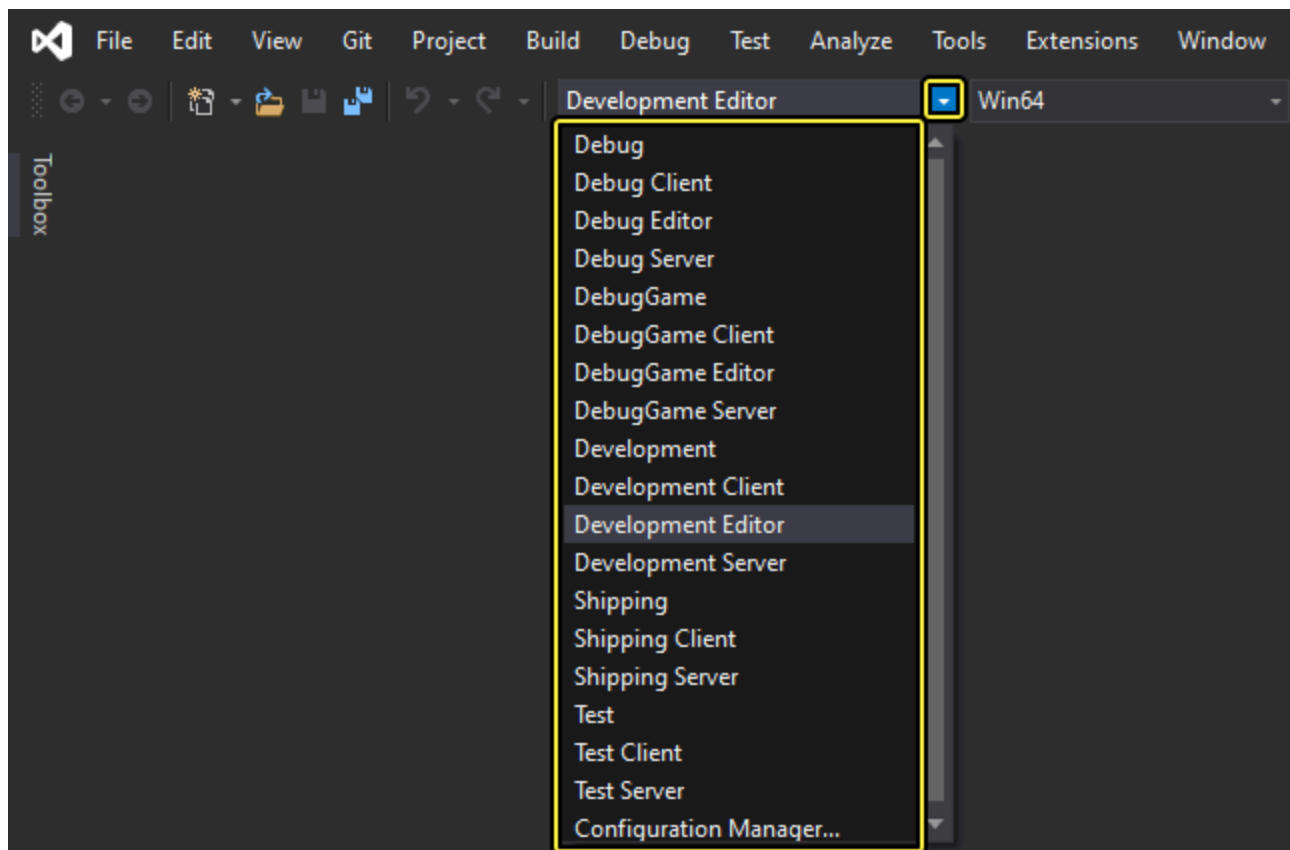
Build Configuration - State	Description
Debug	This configuration builds both engine and game code in debug configuration without optimizations. This makes things slower, but is easier to debug. If you compile your project using the Debug configuration and want to open the project with the Unreal Editor, you must use the <code>-debug</code> flag in order to see your code changes reflected in your project.
DebugGame	This configuration builds game code without optimizations. This configuration is ideal for debugging only game modules.
Development	This configuration enables all but the most time-consuming engine and game code optimizations, which makes it ideal for development and performance reasons. Unreal Editor uses the Development configuration by default. Compiling your project using the Development configuration enables you to see code changes made to your project reflected in the editor.
Shipping	This is the configuration for optimal performance and shipping your game. This configuration strips out console commands, stats, and profiling tools.
Test	This configuration is the Shipping configuration, but with some console commands, stats, and profiling tools enabled.

Build Configuration - Target	Description
Game	This configuration builds a stand-alone executable version of your project, but requires cooked content specific to the platform. Please refer to the Packaging Projects Reference page to learn more about cooked content.
Editor	To open a project in Unreal Editor and see all code changes reflected, the project must be built in an Editor configuration.
Client	If you are working on a multiplayer project using UE networking features, this target designates the specified project as being a Client in UE's client-server model for multiplayer games. If there is a <code><GAME_NAME>Client.Target.cs</code> file, the Client build configurations will be valid.
Server	If you are working on a multiplayer project using UE networking features, this target designates the specified project as being a Server in UE's client-server model for multiplayer games. If there is a <code><GAME_NAME>Server.Target.cs</code> file, the Server build configurations will be valid.

Building with Visual Studio

Setting the Build Configuration

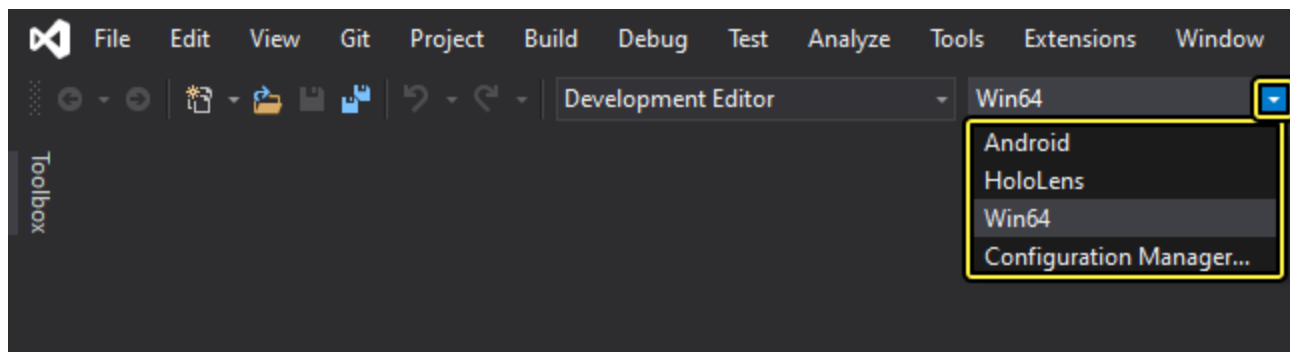
The build configuration can be set in the Visual Studio toolbar. This settings you can find as following.



Setting the Solution Platform

The solution platform can be set in the Visual Studio toolbar.

When working with Unreal Engine, you will typically use the **Win64** platform. This is the only one included by default when generating project files; the [Project Files for IDEs](#) page has instructions for generating project files for additional platforms.

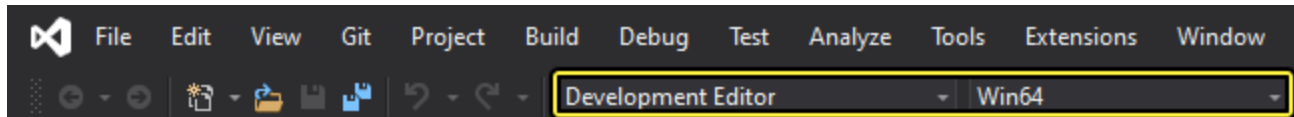


Building the Project

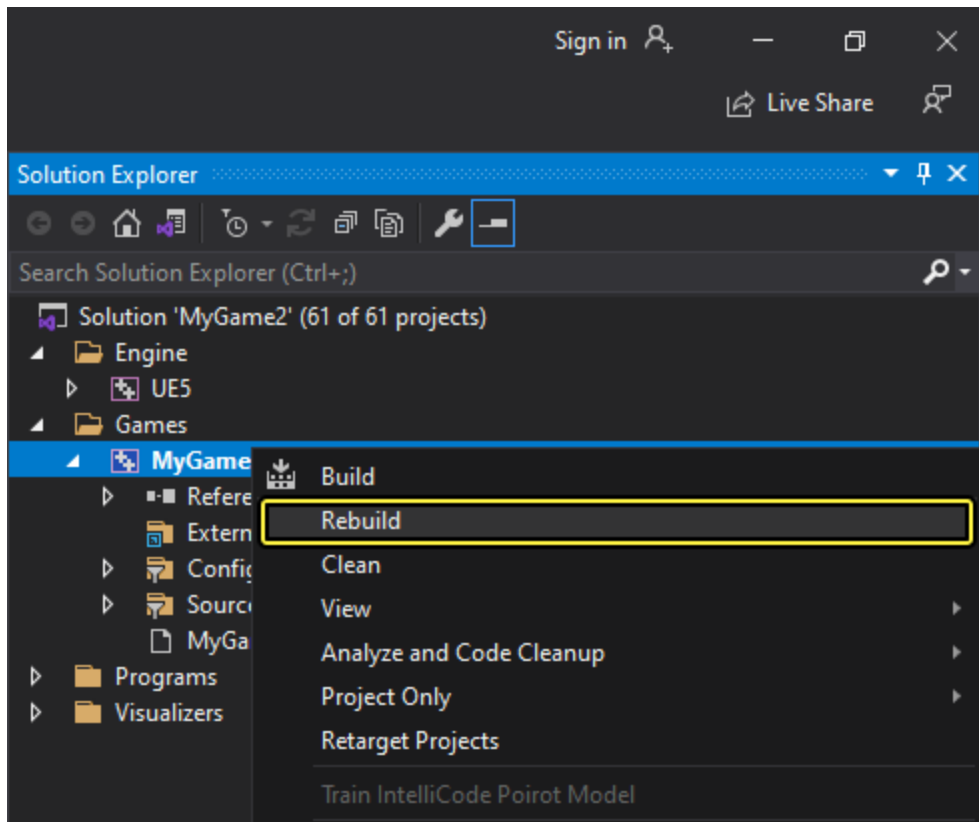


Make sure you are running Visual Studio 2019 or higher for Windows Desktop installed before proceeding. If you are using Mac, make sure to have Xcode 9.0 or higher installed.

1. Set the **Solution Configuration** to the configuration you want to build. In this example, it is set to **Development Editor**. Refer to the [Build Configuration](#) section for descriptions of each available configuration.



2. Right-click your game project and choose **Rebuild** to recompile.



You can now [run Unreal Engine](#) with your compiled project.



When running UE, it is important to use the Unreal Engine executable that matches the build configuration you rebuilt your project in. For example, if you compiled your project in the **DebugGame Uncooked** build configuration, you would run the `UnrealEditor-Win64-DebugGame.exe` executable with your game information as an argument. More information on the binary naming convention can be found on the [Building Unreal Engine](#) page.



When running Unreal Engine, it is important to add the `-game` flag if you rebuilt your project in any **Uncooked** configuration, and the `-debug` flag if you rebuilt your project in any **Debug** configuration.

Visual Studio Known Issues

Issue	Solution
"Project is out of date" message always appears	Visual Studio thinks the project is out of date when it really is up to date. You can safely suppress this message by checking the Do not show this dialog again box and pressing No .
No Debugging Information window appears when using the Debug configuration	The reason for this is that the <code>UnrealEditor.exe</code> was compiled using the Development configuration. Your game project will still be debuggable when compiled using the Debug configuration despite this warning. You can safely suppress this message by checking the Do not show this dialog again box and pressing Yes .