# Visual Studio Integration Tool Documentation

## Description

The Visual Studio Integration Tool is a plugin that enables deep integration with Visual Studio 2022. This tool is required to display Blueprints information in Visual Studio.

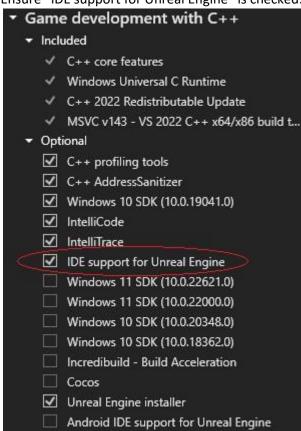
# Requirements

- Visual Studio 2022 17.4
- Unreal Engine 4.27 or 5.0

## Installation

#### Visual Studio 2022

- 1. Locate "Game Development for C++" workload in the Visual Studio Installer.
- 2. Ensure "IDE support for Unreal Engine" is checked.



### **Unreal Engine**

#### Installing via Unreal Engine Marketplace

- 1. In the Epic Games Launcher, select the Unreal Engine Tab
- 2. Click Marketplace
- 3. Search for "Visual Studio Integration Tool"
- 4. Select the plugin, then click Add to Cart
- 5. Click the Cart button and check out
  - a. Note: Visual Studio Integration Tool is free
- 6. Go to the Visual Studio Integration Tool plugin page in the Epic Games Launcher
  - a. You can find this page by searching for the plug in, then clicking the tile
- 7. On the Visual Studio Integration Tool plugin page, click Install to Engine
- 8. Select the engine version you want to install to, then click **Install**
- 9. After the installation is complete, open the version of Unreal Engine you installed the plugin for, and **enable** the plugin following the instructions in the Enabling a Plugin section on this page.
- 10. See <u>official documentation</u> for more information.

## Usage

## **Blueprints Integration**

- In Visual Studio, any Blueprints references will now show up as a <u>CodeLens</u>.
- Click the CodeLens to display a pop-up that shows your Blueprint information (figure 1).
- The Output Window will display the Unreal Engine logs from the plugin execution.
- To refresh the blueprint information in the IDE (e.g., after making asset changes in the Unreal Editor, or fixing an error in the plugin execution), you can use the "Rescan UE Blueprint for cproject>" option under the "Project" menu (figure 2).

```
Blueprint
                                 Path
      BP_PlayerCharacter
                                 /Game/Blueprints/BP_PlayerCharacter.BP_PlayerCharacter
      BP_EnemyCharacter
                                 /Game/Blueprints/BP_EnemyCharacter.BP_EnemyCharacter
      BP_Character
                                 /Game/Blueprints/BP_Character.BP_Character
      NPC_Goblin_Level_03
                                 /Game/Blueprints/NPC/NPC_Goblin_Level_03.NPC_Goblin_Level_03
      NPC_Goblin_Level_02
                                 /Game/Blueprints/NPC/NPC_Goblin_Level_02.NPC_Goblin_Level_02
      NPC_Goblin_Level_01
                                 /Game/Blueprints/NPC/NPC_Goblin_Level_01.NPC_Goblin_Level_01
      NPC_GoblinBP
                                 /Game/Blueprints/NPC/NPC_GoblinBP.NPC_GoblinBP
       UCLASS 8 Blueprint references
      class ACTIONRPG_API ARPGCharacterBase : public ACharacter, public IAbilitySystemInterface, p
            GENERATED_BODY();
24
            /** Returns maximum health, health will never be greater than this */
            UFUNCTION(BlueprintCallable)
            virtual float GetHealth() const;
            /** The level of this character, should not be modified directly once it has already spa
            UPROPERTY(EditAnywhere, Replicated, Category = Abilities)
            int32 CharacterLevel;
            /** Returns maximum health, health will never be greater than this */
            UFUNCTION(BlueprintCallable)
```

Figure 1 - Code Lens showing blueprints that derive from a UEClass

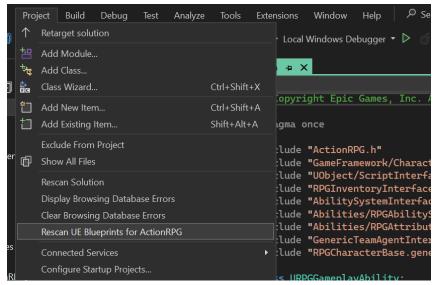


Figure 2 - Menu to rescan the blueprint assets in the game project