

- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Programming and Scripting
- / Blueprints Visual Scripting
- / Specialized Blueprint Node Groups
- / Types of Blueprints
- / Blueprint Class

# Blueprint Class

These define a new class or type of Actor which can then be placed as instances that behave like any other type of Actor.



A **Blueprint Class**, often shortened as **Blueprint**, is an asset that allows content creators to easily add functionality on top of existing gameplay classes. Blueprints are created inside of Unreal Editor visually, instead of by typing code, and saved as assets in a content package. These essentially define a new class or type of Actor which can then be placed into maps as instances that behave like any other type of Actor.

## Parent Classes

There are several different types of Blueprints that you can create, however before doing so you will need to specify the **Parent Class** in which the Blueprint will be based. Selecting a Parent Class allows you to inherit properties from the Parent to use in the Blueprint you are creating.

Below are the most common Parent Classes used when creating a new Blueprint:

Class Type	Description
<b>Actor</b>	An Actor is an object that can be placed or spawned in the world.
<b>Pawn</b>	A Pawn is an Actor that can be "possessed" and receive input from a Controller.
<b>Character</b>	A Character is a Pawn that includes the ability to walk, run, jump, and more.
<b>PlayerController</b>	A Player Controller is an Actor responsible for controlling a Pawn used by the player.
<b>Game Mode</b>	A Game Mode defines the game being played, its rules, scoring, and other faces of the game type.

While the above are the most common, all existing classes can be used as the Parent Class for a new Blueprint (even other Blueprint Classes).

For example, say you created an **Actor Blueprint** called *Animals* and in it provided some script that all animals share such as *Hunger*, *Thirst*, *Energy*, or whatever script you wanted. Then you created another Blueprint called *Dogs* and selected your *Animals* Blueprint Class as the Parent Class; you can then provide specific functionality that applies to only dogs within the *Dogs* Blueprint such as *Play Fetch* or *Roll Over* while inheriting the functionality that all animals share from the *Animals* Blueprint.



Depending upon the method used to [create a Blueprint Class](#), a Parent Class may already be assigned upon creation.

## Working with Blueprint Classes

Refer to the sections below for information on working with **Blueprint Classes**:



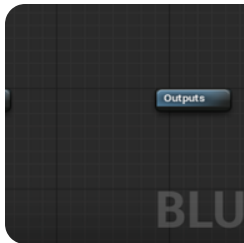
## Blueprint Editor Blueprint Class UI

A breakdown of the UI elements of the Blueprint Editor when working on Blueprint Classes.



## Creating Blueprint Classes

A how-to guide on ways to create new Blueprint classes



## Blueprint Editor Reference

The Blueprint Editor Reference page outlines the Blueprint Editor's Interface elements and its basic usage instructions.

# Data-Only Blueprint

A **Data-Only Blueprint** is a Blueprint Class that contains only the code (in the form of node graphs), variables, and components inherited from its parent. These allow those inherited properties to be tweaked and modified, but no new elements can be added. These are essentially a replacement for archetypes and can be used to allow designers to tweak properties or set items with variations.

Data-Only Blueprint are edited in a compact property editor, but can also be "converted" to full Blueprints by simply adding code, variables, or components using the full **Blueprint Editor**.