Actors Reference

Describes the most common types of Actors in Unreal Engine and where you can learn more about them.

This page describes the most common types of Actors you will be working with in Unreal Engine. The Actors are grouped by purpose and functionality. Each section includes links to other pages and resources where you can learn more about these Actors and what you can do with them.

Note that this is not a comprehensive list of every Actor type available in Unreal Engine. Some plugins and project templates add their own Actors, and certain Actors may not be available for all projects.

Mesh Actors

A **mesh** defines the shape and size of an environment prop or a player character. Unreal Engine uses two types of mesh Actors:

- Static Mesh Actors, which are used to build levels and environments.
- Skeletal Mesh Actors, which are typically used for player characters and animated non-player characters (NPCs).

Static Mesh Actors

The **Static Mesh Actor** is a simple type of Actor that displays a mesh in the Level. Although the name implies that the Actor is static (or unable to move), the "static" refers to the type of mesh used rather than the Actor's ability to move. A mesh is *static* if its geometry does not change. Otherwise, the Actor itself can be moved or changed in other ways during play.

Static Mesh Actors are commonly used to create game worlds or other types of environments.

Refer to the Static Meshes section to learn more.

Skeletal Mesh Actors

The **Skeletal Mesh Actor** displays an animated mesh whose geometry can be deformed, typically through the use of control points during animation sequences. These can either be created and exported from external 3D animation applications, or programmed directly in Unreal Engine.

Skeletal Mesh Actors are commonly used to represent player characters or NPCs, as well as other animated creatures and complex machinery.

Refer to the Skeletal Mesh Actors page to learn more.

Brush Actors (Geometry Actors)

Brush Actors are a basic type of Actor that displays simple 3D geometry in the scene, such as spheres, cubes, and stairs. These Actors can be changed using the Geometry Editing mode in the Level Editor.

The following types of Brush Actors are available in Unreal Engine:

- Box
- Cone
- Cylinder
- Curved Stair
- Linear Stair
- Spiral Stair
- Sphere

Brush Actors are commonly used to quickly prototype environments and block out levels. Use these to get a rough impression of the size and placement of environment props.

Refer to the Geometry Brush Actors page to learn more.

Light Actors

As the name suggests, **Light Actors** are used to place different types of lights around your Level. Some Light Actors have a limited area of effect around them, while others affect your entire Level.

The following types of Light Actors are available in Unreal Engine:

- Directional Light
- Point Light
- Spot Light
- Rect Light
- Sky Light

You can learn more about lighting fundamentals in Unreal Engine from the following Unreal Online Learning courses:

- Lighting Essential Concepts and Effects
- Introducing Global Illumination

Directional Light Actor

A **Directional Light** simulates light that is being emitted from a source that is infinitely far away. This means that all shadows cast by this light will be parallel, making this the ideal choice for simulating sunlight.

Refer to the Directional Lights page to learn more.

Point Light Actor

Point Lights work like real-world light bulbs. A Point Light emits light in all directions from its center, which is a single point in the Level.

Refer to the Point Lights page to learn more.

Spot Light Actor

Spot Lights work like a stage light or a flashlight. A Spot Light emits light outward in a conical shape from a single point. The Spot Light's shape is defined by two separate cone angles:

- Inner Cone Angle, inside which the light achieves full brightness.
- Outer Cone Angle, which defines the outer bounds of the Spot Light.

From the Inner Cone Angle and extending towards the limits of the Outer Cone Angle, the light's intensity falls off, softening around the Spot Light's disc of illumination.

Refer to the Spot Lights page to learn more.

Rect Light Actor

Rect Lights emit light into the Level from a rectangular plane with a defined width and height. You can use these to simulate any kind of light sources that have rectangular surface areas, such as windows, televisions, or monitor screens.

Refer to the Rectangular Area Lights page to learn more.

Sky Light Actor

The **Sky Light** captures distant parts of your Level and applies them to the scene as a light source. That means the sky's appearance and its lighting / reflections will match, whether your sky is coming from the atmosphere, layered clouds on top of a skybox, or from distant mountains.

If you are developing an Augmented Reality (AR) application, consider using an **ARSky Light** Actor instead.

ARSkyLight is a subclass of the SkyLight class that uses real-world environment probes to update reflections. It regenerates lighting and reflections whenever the corresponding probe's texture updates based on the real-world lighting. This all occurs on the render thread.

Refer to the Sky Lights page to learn more.

Camera Actors

Like their real-world counterpart, **Camera Actors** are used to view your Level and create cinematic sequences. In addition, there are a number of supporting Actors that can be used to simulate real-world camera shots.

Refer to the Camera Actors page to learn more.

Audio and Sound Actors

Audio and Sound Actors are used to add music, voice recordings, and sound effects to your Level.

Ambient Sound Actor

Use an **Ambient Sound Actor** to play a looping (continuous) sound at a specific location in the Level.

Refer to the Ambient Sound Actor User Guide page to learn more.

Audio Volume

With the **Audio Volume Actor**, you can define areas within a Blueprint graph that can be used to process sounds, and use its settings to apply reverb effects, set volumes, define the zones affected, emulate occlusion on a sound, and define the shape of the sound volume.

Refer to the Audio Volume page to learn more.

Gameplay Actors

Gameplay Actors trigger interactive functionality. Despite their name, they have a broad range of uses in all kinds of interactive applications, not just games.

Below are the most common types of Gameplay Actors.

> Anything that you can place in your Level is called an Actor. A Volume is a type of Actor that has a three-dimensional shape, such as a cube or sphere.

Player Start

A **Player Start** is an Actor that is placed in the Level to designate where the player character spawns when the player starts the Level.

Refer to the Player Start Actor page to learn more.

Trigger Volumes

Triggers are Actors that cause an event to occur when something else in the Level interacts with them, such as the player character or another object. For example, a player can interact with a trigger to switch on a light.

The following types of triggers are available in Unreal Engine:

- Box Trigger
- Capsule Trigger
- Sphere Trigger

All three of these trigger types have the same functionality. The only thing that's different is the shape. You can choose between different shapes depending on the kind of visual representation the trigger has in your Level.

In addition, Unreal Engine also has a generic Trigger Volume. Much like Geometry Actors, you can use this volume to quickly prototype interactions in your Level.

Refer to the Trigger Volume Actors page to learn more.

Blocking Volume

As the name suggests, **Blocking Volumes** are used to prevent the player from passing through. For example, you might use Blocking Volumes so your player doesn't fall off the edges of your game world.

Kill ZVolume

The **Kill ZVolume Actor** instantly "kills" (destroys) the player character once they enter or interact with its volume. You can specify the kill condition in the Kill ZVolume Actor's Details panel.

> ⓘ Note that the Kill ZVolume Actor is different from the Kill Z option in the **World Settings** panel, which destroys the player once they fall past a certain point along the Z axis (vertical axis).

Pain Causing Volume

A **Pain Causing Volume** causes damage over time to any player or object that enters it. For example, if a player stands in the fire, you can set up a Pain Causing Volume to deal corresponding damage to their health.

> Don't stand in the fire.

Refer to the Pain-Causing Volume Actor page to learn more.

Character and Pawn Actors

Both **Pawns** and **Characters** are Actors that represent player and AI-controlled characters.

Pawn

A **Pawn** is the physical representation of a player or AI entity within the world. The Pawn not only determines what the player or AI entity looks like visually, but also how it interacts with the world in terms of collisions and other physical interactions.

Depending on the kind of project you're building, the Pawn could be a player avatar, a car, or have no physical representation at all.

There are two types of Pawns available in Unreal Engine 5:

Pawn Type	Description
Pawn	This is an empty Pawn Actor. It doesn't have any visual representation (mesh) or controls attached to it.
Default Pawn	This is a simple Pawn Actor with spherical collision and built-in flying movement. Note that you need to attach a controller Component to it before it responds to player input.

Character

A **Character** is a specific type of Pawn that is designed for a vertically-oriented player character that can walk, run, jump, fly, and swim through the world. In other words, if your player controls a bipedal avatar (for instance, a human), the avatar will be a Character rather than a Pawn.

There are several types of Character Actors available in Unreal Engine. When you set up your project, choose the one that best matches the project type.

Character Type	Description	Further Reading
Character	This is a simple Character Actor with capsule collision, an empty mesh, and character movement. As-is, this Actor is invisible in the Level, but can be controlled.	
Arch Vis Character	The Arch Vis Character is based on Unreal Engine's First Person Character class. Some of its parameters, such as warp and turn speed, have been adjusted to allow for smoother movement. This Character is optimized for architectural visualization projects.	Using the Archviz Character Pawn (YouTube)
Paper Character	The Paper Character is used for 2D projects. The difference between a Paper Character and a generic Character is that the Paper Character has a Paper Flipbook Component (a Sprite) to represent the character visually, whereas the generic Character uses a three-dimensional mesh.	Paper 2D Sprites

Visual Effects Actors

Visual Effects Actors are used to change the look and feel of your Level. These Actors only have a limited area of effect, defined by a finite three-dimensional volume.

Post Process Volume

The **Post Process Volume** applies one or more visual effects to the objects contained within it. Unreal Engine offers a broad range of effects that can be applied, from bloom and vignettes to global illumination and ray tracing reflections.

ⓘ Depending on the size and complexity of your Level, as well as your device's specs, post-processing can be a resource-intensive process and can have a significant impact on performance at runtime.

Refer to the Post Process Effects section to learn more.

Reflection Capture Actors

Reflection Capture Actors are used to capture the contents of the Level for reflection. They capture a static image of the area around them, and map it to a sphere or a box shape, depending on the Actor's shape. Anything within the volume of a Reflection Capture Actor will have its static image reflected by any reflective surfaces around it.

The following types of Reflection Capture Actors are available in Unreal Engine:

- Sphere Reflection Capture
- Box Reflection Capture

Refer to the Reflections Environment page to learn more.

Planar Reflection Actor

A **Planar Reflection Actor** captures a 2D mirror image of the scene. It is ideal for creating dynamic mirror reflections and for capturing things that aren't within the current camera view.

Refer to the Planar Reflections page to learn more.

Decal Actor

Decal Actors can be placed over the surface of a mesh to render a Material on top of it, much like a real-world "sticker". You can use decals to add details and variations to multiple surfaces that use the same texture, like adding water leaks or paint splatters to modular walls.

Refer to the following pages to learn more:

- Decal Actors
-

Worldbuilding Actors

Use **Worldbuilding Actors** to add realistic details to your Level, such as sky atmosphere, fog, and volumetric clouds.

Sky Atmosphere Actor

The **Sky Atmosphere Actor** is a physically-based sky and atmosphere-rendering technique. It's flexible enough to create an Earth-like atmosphere with time-of-day featuring sunrise and sunset, or to create extraterrestrial atmospheres of an exotic nature. It also provides an aerial perspective to which you can simulate transitions from ground to sky to outer space with proper planetary curvature.

Refer to the Sky Atmosphere page to learn more.

Volumetric Cloud Actor

The **Volumetric Cloud Actor** is a physically-based cloud rendering system that uses a material-driven approach to give artists and designers the freedom to create any type of clouds they need for their projects. The system provides scalable clouds that can adapt to projects using ground views, aerial views, and ground to outer space transitions

Refer to the Volumetric Clouds page to learn more.

Exponential Height Fog Actor

Exponential Height Fog creates more density at lower altitudes within a Level and less density at higher altitudes. The transition is smooth, so you never see a hard cutoff as you increase altitude.

Exponential Height Fog also provides two fog colors — one for the hemisphere facing the dominant directional light (or directly upward if none exists), and another color for the opposite hemisphere.

Refer to the Exponential Height Fog User Guide page to learn more.

Other Actors

Unreal Engine contains a number of other Actors you can use to add various elements to your Level. Some of these Actors are enabled by default, while others require that you enable specific plugins first. To learn how to enable plugins in Unreal Engine, refer to the Working with Plugins page.

Text Render Actor

The **Text Render Actor** provides a simple way to add text to your Level. For a quick usage example, create a new Third Person project. The blue text that reads "THIRD PERSON" across the floor is rendered using a Text Render Actor.

You can create and animate more complex 3D text using the **Text 3D** plugin. Refer to the 3D Text Actor page to learn more.

Target Point Actors

Target Point Actors give you a generic point in the world that you can spawn items from. If you are familiar with other 3D applications such as 3Ds Max or Maya, Target Point Actors are very similar to dummy Actors that you find in those programs.

Refer to the Target Point Actors page to learn more.