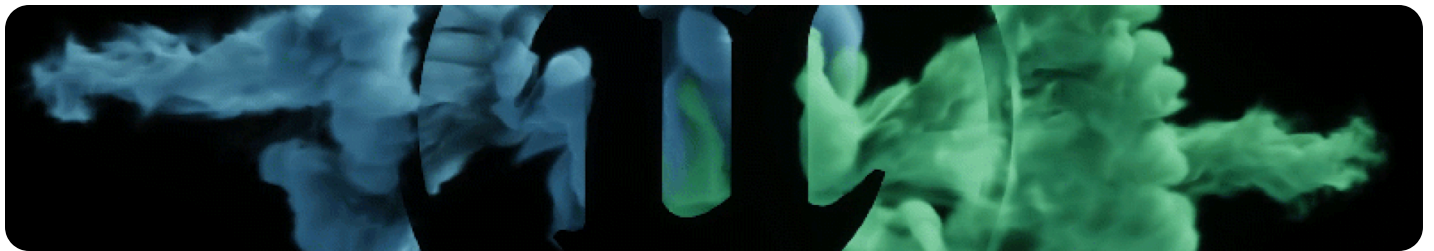


Developer
/ Documentation
/ Unreal Engine ▾
/ Unreal Engine 5.4 Documentation
/ Creating Visual Effects
/ Niagara Lightweight Emitters
/ Niagara Lightweight Emitters Overview

Niagara Lightweight Emitters Overview

Overview of Niagara's Lightweight Emitters.



Overview

Unreal Engine 5.4 introduces Niagara **lightweight emitters**. These emitters, also called **stateless emitters**, are optimized to minimize (and in some cases eliminate) the use of tick during the simulation.

In addition, lightweight emitters don't need to be compiled, which results in a faster workflow and the amount of emitters in a system will have less impact on performance. This means that in most cases a stateless emitter will be significantly more performant than stateful (traditional) emitter.

You add a stateless emitter inside a Niagara System by right clicking inside the System Overview window and selecting **Add stateless emitter** from the menu. Niagara Systems can have both types of emitters working concurrently, but a pure stateless system will be the most performant.

Improved Performance and Faster Development

The primary motivation for introducing stateless emitters was to reduce Niagara's memory and CPU cost, and to accelerate development time for VFX artists.

To achieve this, stateless emitters do the following:

- **Reduce Game Thread tick cost** if the Niagara System is fully stateless.
- **Remove the concurrent tick cost** per stateless emitter.
- **Reduce Render Thread cost** when the Niagara System is fully stateless.
- **Reduce memory cost**, as there are no scripts or particle information in memory.
- Removes **performance impact** of **emitter count** and **particle instance count**.
- **Reduce** (and sometimes remove) the need for **compilation**.

Stateless Emitter Trade-offs

To achieve improved performance, stateless emitters are fixed-function and only have access to the following modules:

- Acceleration Force
- Add Velocity
- Curl Noise / Noise Vector Field
- Drag
- Gravity Force
- Initialize Particle
- Initial Mesh Orientation
- Rotate Around Point
- Scale Color
- Scale Mesh Size
- Scale Mesh Size By Speed
- Scale Sprite Size
- Scale Sprite Size By Speed
- Shape Location
- Solve Velocities And Forces
- Sprite Rotation Rate
- Sub UV Animation

You can adjust the settings for each of the available modules, but cannot create custom modules, scratchpads, or use dynamic inputs. The current feature-set can be expanded via C++ and you can convert a stateless emitter to a stateful, or vice versa, inside your Niagara System.