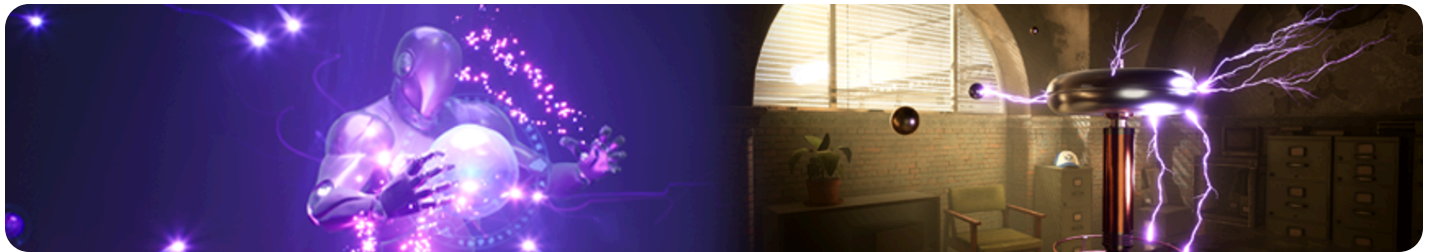


- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Creating Visual Effects
- / Niagara Reference
- / Niagara System and Emitter Module Reference
- / Emitter Spawn Group

# Emitter Spawn Group

This document provides reference information for modules in the Emitter Spawn group.



**Emitter Spawn** modules occur once, when the emitter is created on the CPU. Modules in this section should initialize defaults or perform initial setup. Modules are executed in order from the top to the bottom of the stack.

Each of the module types in the Emitter Spawn group has its own section in this document, with tables that list and describe the default options available for that type of module. Keep in mind that you can create custom modules for any part of a Niagara emitter. The ones listed here are just the ones that are automatically included with Unreal Engine 4.

## Location Module

| Module                         | Description   |
|--------------------------------|---|
| <b>Spawn Particles in Grid</b> | This spawns particles based on user-defined grid resolution settings. |

# MAX Scripts Module

| Module   | Description  |
|--|--|
| <b>Spawn MS Vertex Animation Tools Morph Target</b>  | This spawns and samples Morph Target textures that were created with the Vertex Animation Tool. The Vertex Animation Tool generates textures that represent Morph Target blend shapes. This module spawns one particle for every vertex that was captured by the tool. This module is meant to be used with the <b>Update MS Vertex Animation Tools Morph Target</b> module. |
| <b>Update MS Vertex Animation Tools Morph Target</b> | This takes the particles spawned by the <b>Spawn MS Vertex Animation Tools Morph Target</b> module and places them at a location you choose.   |

## Spawning Modules

| Module                | Description  |
|-----------------------|--|
| <b>Spawn Per Unit</b> | This module spawns particles based on distance traveled in Unreal units. |
| <b>Spawn Rate</b>     | This module spawns particles continuously at a particular rate.          |

## New Scratch Pad Module

Selecting this item in the **Add** (Plus sign) menu opens the **Scratch Pad** panel (by default this docks next to the **System Overview**) and places a **Scratch Pad module** in the **Selection** panel. You can also open the Scratch Pad panel by using **Windows > Scratch Pad**. However, by placing a Scratch Pad module in the stack, any modules or dynamic inputs you create in the Scratch Pad are automatically connected to your script. If you open the Scratch Pad panel using the Windows menu, any items you create there will have to be added to your script manually.

# Set New or Existing Value Directly

Selecting this item from the **Add** menu places a **Set Parameter** module in the **Selection** panel. Click the **Plus sign (+)** icon to select **Add Parameter** or **Create New Parameter**.

## Add Parameter

When you select **Add Parameter**, you select from the parameters listed. This adds that parameter to the **Set Parameter** module in the Emitter Spawn group.



Some of these parameters can be set or modified in other modules. Some are only set using a Set Parameter module.

| Parameter                           | Description  |
|-------------------------------------|--|
| <b>Emitter.Age</b>                  | This parameter defines the age of this emitter.  |
| <b>Emitter.CurrentLoopDelay</b>     | This parameter defines the current amount of delay before the emitter's current loop repeats.  |
| <b>Emitter.CurrentLoopDuration</b>  | This parameter defines the duration of the current emitter loop.   |
| <b>Emitter.ExecutionState</b>       | This affects the state of the emitter. Valid value choices are: <ul style="list-style-type: none"><li>• <b>Active</b></li><li>• <b>Inactive</b></li><li>• <b>InactiveClear</b></li><li>• <b>Complete</b></li></ul> |
| <b>Emitter.ExecutionStateSource</b> | This indicates the source of an execution state setting. It is used to allow scalability to change the state, but only if the state has not been defined by something with higher precedence.                      |

| Parameter                          | Description   |
|------------------------------------|---|
| <b>Emitter.LocalSpace</b>          | <p>This parameter defines whether the position of particles is respective to the world origin or the owning Niagara Component's location.</p> <ul style="list-style-type: none"> <li>• <b>False:</b> Particle position is in WorldSpace and will be relative to the World origin. A particle with position <b>0,0,0</b> will render at the World origin.</li> <li>• <b>True:</b> Particle position is in LocalSpace and will be relative to the owning Niagara Component's location. A particle with position <b>0,0,0</b> will render at the owning Niagara Component's location.</li> </ul> |
| <b>Emitter.LoopCount</b>           | This parameter defines how many times the emitter's loop repeats.   |
| <b>Emitter.LoopedAge</b>           | This parameter calculates the age of the emitter relative to its current loop. For example, if an emitter has been active for 8 seconds and it loops every 5 seconds, the emitter's LoopedAge will be <b>3 seconds</b> . LoopedAge returns to 0 every time an emitter loops.  |
| <b>Emitter.NormalizedLoopAge</b>   | This parameter calculates the age of the emitter relative to its current loop, normalized from <b>0</b> to <b>1</b> . NormalizedLoopAge is expressed as <b>LoopedAge</b> divided by <b>CurrentLoopDuration</b> . If an emitter has been active for 8 seconds and it loops every 5 seconds, the emitter's LoopedAge will be <b>3</b> . The emitter's NormalizedLoopAge will be <b>0.6</b> .  |
| <b>System.ExecutionState</b>       | This affects the state of the system. Valid value choices are <b>Active</b> , <b>Inactive</b> , <b>InactiveClear</b> , <b>Complete</b> , <b>Disabled</b> , and <b>Num</b> .   |
| <b>System.ExecutionStateSource</b> | This indicates the source of a system execution state setting. It is used to allow scalability to change the state, but only if the state has not been defined by something with higher precedence.   |

# Create New Parameter

When you select **Create New Parameter**, you select from the parameters listed. This adds that parameter to the **Set Parameter** module in the Emitter Spawn group.


| Parameter                 | Type                  | Description   |
|---------------------------|-----------------------|---|
| <b>Audio Oscilloscope</b> | <b>Data interface</b> | This adds a new Audio Oscilloscope data interface module to the emitter. The Audio Oscilloscope module can directly access the waveform data of the audio signal.   |
| <b>Audio Spectrum</b>     | <b>Data interface</b> | This adds a new Audio Spectrum data interface module to the emitter. The Audio Spectrum module can drive a visualization based on how loud the audio is at specific frequencies.                            |
| <b>Bool</b>               | <b>Primitive</b>      | This adds a Set Variable module that has a true/false checkbox.   |
| <b>Camera Query</b>       | <b>Data Interface</b> | This adds a new Camera Query data interface module to the emitter. This data interface can be used to retrieve camera information (camera position, rotation, FOV, etc) for the specified controller index. |
| <b>Collision Query</b>    | <b>Data Interface</b> | This adds a collision data interface to the emitter stack. This is usually used in conjunction with Collision modules.  |


| Parameter                   | Type                  | Description  |
|-----------------------------|-----------------------|--|
| <b>Curl Noise</b>           | <b>Data Interface</b> | This adds a curl noise data interface to the emitter stack. If you use this in conjunction with Curl Noise Force modules, this data interface injects different types of noise into your simulation. |
| <b>Curve for Colors</b>     | <b>Data Interface</b> | This adds a four-channel color curve data interface for the simulation. This curve can be sampled by dynamic inputs or other modules to create a time-varying color.                                 |
| <b>Curve for Floats</b>     | <b>Data Interface</b> | This adds a single-channel curve data interface for the simulation. This curve can be sampled by dynamic inputs or other modules to create a time-varying float value.                               |
| <b>Curve for Vector 2Ds</b> | <b>Data Interface</b> | This adds a two-channel curve data interface for the simulation. This curve can be sampled by dynamic inputs or other modules to create a time-varying pair of floats.                               |
| <b>Curve for Vector 3s</b>  | <b>Data Interface</b> | This adds a three-channel curve data interface for the simulation. This curve can be sampled by dynamic inputs or other modules to create a time-varying set of floats.                              |
| <b>Curve for Vector 4s</b>  | <b>Data Interface</b> | This adds a four-channel curve data interface for the simulation. This curve can be sampled by dynamic inputs or other modules to create a time-varying set of floats.                               |

| Parameter                           | Type        | Description   |
|-------------------------------------|-------------|---|
| <b>ENiagaraBooleanLogicOps</b>      | <b>Enum</b> | <p>This is an enumeration used by various modules and dynamic inputs that want to test using boolean logic:</p> <ul style="list-style-type: none"> <li>• <b>Greater Than</b></li> <li>• <b>Greater Than Or Equal To</b></li> <li>• <b>Equal To</b></li> <li>• <b>Not Equal To</b></li> </ul>  |
| <b>ENiagaraCoordinateSpace</b>      | <b>Enum</b> | <p>This is an enumeration used by various modules and dynamic inputs to distinguish between coordinate spaces:</p> <ul style="list-style-type: none"> <li>• <b>Simulation:</b> If the emitter is set to local, use local. Otherwise, use World.</li> <li>• <b>World:</b> In the world space of the game.</li> <li>• <b>Local:</b> In the coordinate space of the owning component.</li> </ul> |
| <b>ENiagaraExecutionState</b>       | <b>Enum</b> | <p>This enumeration type is used by parameters that manage system or emitter execution states, such as <b>Emitter.ExecutionState</b> or <b>System.ExecutionState</b>.</p>   |
| <b>ENiagaraExecutionStateSource</b> | <b>Enum</b> | <p>This indicates the source of an execution state setting. It is used to allow scalability to change the state, but only if the state has not been defined by something with higher precedence.</p>  |
| <b>ENiagara ExpansionMode</b>       | <b>Enum</b> | <p>This enumeration is used by location modules to determine where the origin point of expansion is:</p> <ul style="list-style-type: none"> <li>• <b>Inside</b></li> </ul>  |

| Parameter                      | Type                  | Description   |
|--------------------------------|-----------------------|---|
|                                |                       | <ul style="list-style-type: none"> <li>• <b>Centered</b></li> <li>• <b>Outside</b></li> </ul>   |
| <b>ENiagaraOrientationAxis</b> | <b>Enum</b>           | <p>This is an enumeration used by several modules to determine which axis to do calculations with:</p> <ul style="list-style-type: none"> <li>• <b>X Axis</b></li> <li>• <b>Y Axis</b></li> <li>• <b>Z Axis</b></li> </ul>                        |
| <b>ENiagaraRandomnessMode</b>  | <b>Enum</b>           | <p>This sets the type of random number generation used by this emitter. Valid choices are:</p> <ul style="list-style-type: none"> <li>• <b>Simulation Defaults</b></li> <li>• <b>Deterministic</b></li> <li>• <b>Non-Deterministic</b></li> </ul> |
| <b>Float</b>                   | <b>Primitive</b>      | This creates a float value variable.  |
| <b>Grid2D Collection</b>       | <b>Data Interface</b> | This is used with simulation stages. It enables the user to read or write to a 2D array of data, and then iterate over each pixel in the grid during a simulation stage.  |
| <b>Int32</b>                   | <b>Primitive</b>      | This creates an integer variable.   |
| <b>Linear Color</b>            | <b>Primitive</b>      | This creates an RGBA color variable, represented as a color picker.   |
| <b>Matrix</b>                  | <b>Primitive</b>      | This creates a 4×4 matrix variable.   |
| <b>Mesh Tri Coordinate</b>     | <b>Struct</b>         | This is a simple struct containing a triangle index along with a barycentric  |



| Parameter                        | Type                  | Description   |
|----------------------------------|-----------------------|---|
|                                  |                       | coordinate on the face of that triangle.  |
| <b>Neighbor Grid 3D</b>          | <b>Data Interface</b> | This is used with simulation stages. It enables the user to read or write to a 3D array of data, and then iterate over each pixel in the volume during a simulation stage.  |
| <b>Niagara ID</b>                | <b>Struct</b>         | <p>This is a two-part struct used to track particles. It allows fast access to this particle's data. It is always unique among currently living particles, but will be reused after the particle dies.</p> <p><b>AcquireTag</b> is a unique tag for when this ID was acquired. It allows us to differentiate between particles when one dies and another particle reuses the dead particle's index.</p> |
| <b>Occlusion Query</b>           | <b>Data Interface</b> | <p>This adds a new Occlusion Query data interface module to the emitter. The data interface is used to read depth buffer occlusion information.</p> <div>  <p>This can only be used with GPU emitters.</p> </div>   |
| <b>Particle Attribute Reader</b> | <b>Data Interface</b> | This adds a new Particle Attribute Reader data interface to the emitter. The data interface can be used to query particle payload values from other emitters, and can sometimes be easier to use than Events.   |

| Parameter             | Type                  | Description   |
|-----------------------|-----------------------|---|
| <b>Quat</b>           | <b>Primitive</b>      | This creates a quaternion variable, used to represent rotations.  |
| <b>Simple Counter</b> | <b>Data Interface</b> | <p>This adds a new Simple Counter data interface module to the emitter. This data interface enables you to increment a thread-safe counter.</p> <div>  <p>This can only be used with CPU emitter.</p> </div>  |
| <b>Skeletal Mesh</b>  | <b>Data Interface</b> | This is a data interface with functions to interact with a skeletal mesh's bones or sockets and skinned geometry.   |
| <b>Spawn Info</b>     | <b>Struct</b>         | This is a structure used in spawning to specify the <b>Count</b> of particles to create, <b>InterpStartDt</b> or offset from the current frame begin time to start spawning, <b>IntervalDt</b> defining the time gap between particles being spawned, and <b>SpawnGroup</b> allowing spawned particles to belong to different categories. |
| <b>Spline</b>         | <b>Data Interface</b> | This is a data interface that interacts with a Spline Asset.  |
| <b>Static Mesh</b>    | <b>Data Interface</b> | This is a data interface with functions to interact with a static mesh's surface.   |
| <b>Texture Sample</b> | <b>Data Interface</b> | This is a data interface with functions to interact with a texture on the GPU.  |

| Parameter                    | Type                  | Description   |
|------------------------------|-----------------------|---|
| <b>Vector</b>                | <b>Primitive</b>      | This creates a three-channel set of floats.   |
| <b>Vector 2D</b>             | <b>Primitive</b>      | This creates a two-channel set of floats.   |
| <b>Vector 4</b>              | <b>Primitive</b>      | This creates a four-channel set of floats.  |
| <b>Vector Field</b>          | <b>Data Interface</b> | This is a DataInterface with functions to query a vector field.   |
| <b>Volume Texture Sample</b> | <b>Data Interface</b> | This adds a new Volume Texture data interface module to the emitter. You can use this to sample a volume texture. |