Groom Caches

An overview of using imported Groom Caches with grooms.

A **Groom Cache** is an extension of a Groom Asset that allows playback of groom animations that are simulated in other digital content creation (DCC) applications and exported to Unreal Engine using Alembic. A Groom Cache Alembic is a Groom Alembic which contains time samples that animate some attributes. The supported attributes that can be animated are vertex position, width, and color.

Types of Groom Caches

There are two types of Groom Cache Alembic files that can be imported into Unreal Engine: Guides Groom Caches and Strands Groom Caches

A **Guides Groom Cache** supports only vertex position animation of the guides. It uses the in-engine strands interpolation to transfer the guides animation to the render strands, requiring that the Groom Asset enable simulation on each of the groom groups. It is suitable for playing back a groom animation that animates only the vertex position where the in-engine interpolation gives satisfactory results. This type of groom cache is more lightweight and performant than using a Strands Groom Cache.

A **Strands Groom Cache** contains the animated attributes for every strand, causing it to consume more memory and bandwidth than a Guides Groom Cache. The benefit is more flexibility because it supports width and color in addition to vertex position, and is higher fidelity since it contains the exact simulation that was computed in the DCC application. This type of groom cache does not use the in-engine interpolation, and does not require enabling simulation on the Groom Asset. A current limitation of a Strands Groom Cache is it must contain vertex position animation, it cannot have just width and color animation by itself.

Strands Groom Caches are useful for playing back groom animations that cannot be simulated in-engine, or where simulation or interpolation does not give satisfactory results. For example, having an animation that shows hair growth where the hair grows longer and thicker.

> (i) Keep in mind that for a Groom Cache to work with a Groom Asset, they must have the same topology, such as the same number of strands, vertices, vertex per strand, and so on. For hair instances where you want to do hair shortening, the number of vertices globally and per-strand must not change, instead it is their position that must change to animate the strand length.

Importing a Groom Cache into Unreal Engine

Since a Groom Cache is a Groom with a time sample, the Groom Cache Alembic is detected as a regular Groom during import. However, the **Groom Import Options** will show a **Groom Cache** section with **Import Groom Cache** enabled.
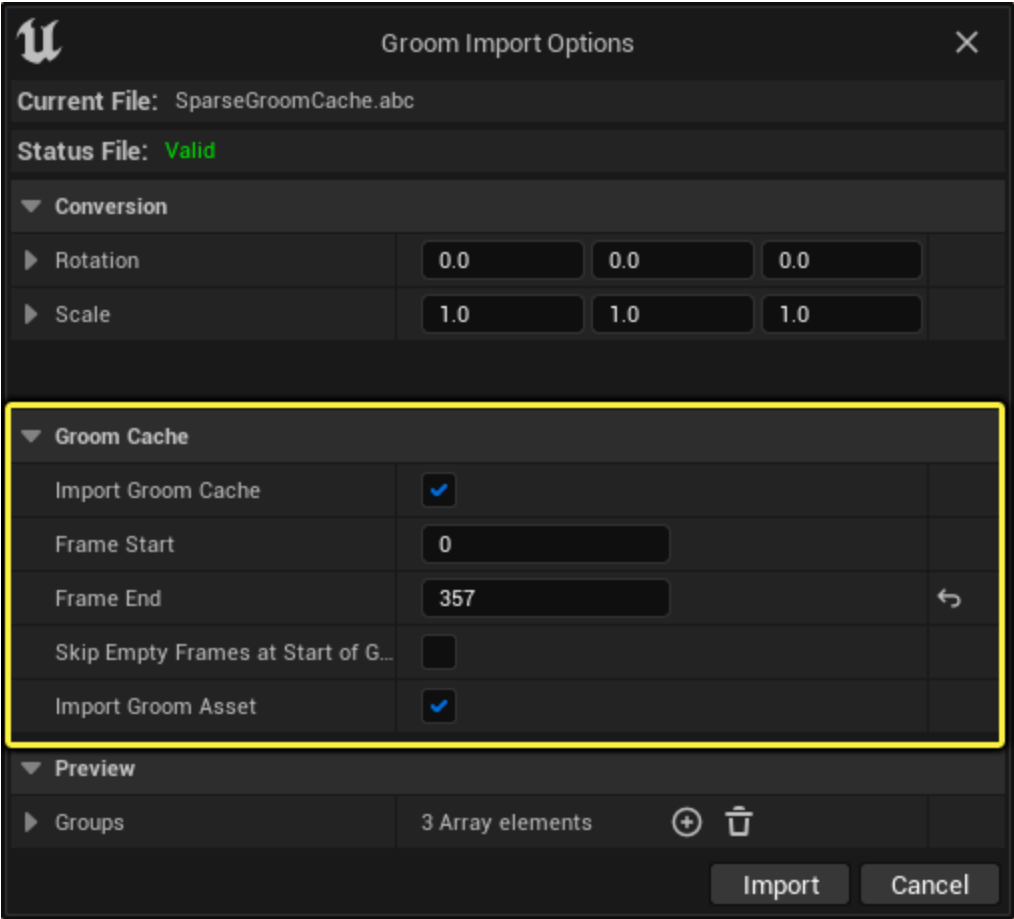
> (i) Import Groom Cache can be disabled to import only the Groom Asset. It is not possible to import a guides-only Groom Asset nor a guides-only Groom Cache.

With a strands-only groom, the guides are automatically generated at import with the interpolation settings, and those settings are used to generate the guides for a Guides Groom Cache. If the guides are already included in the Alembic, they are used instead for both the Groom Asset and the Groom Caches unless the Groom Import Option **Override Guides** is enabled, which is found under the **Groups > Index [n] > Interpolation Settings**.

In cases where you already have a Groom Asset that is topologically compatible with the Groom Cache being imported, you can disable **Import Groom Asset** and specify that Groom Asset to use.

The import process applies the import settings of the specified **Groom Asset** to the Groom Cache you are importing. This is especially important with the decimation and interpolation settings to ensure that the Groom Asset and Groom Caches have the same imported topologies.

> ⓘ When **Import Groom Asset** is disabled, you cannot import the Groom Cache until a **Groom Asset** is specified.

The other settings in the Groom Cache section are there to control the imported animation range, such as specifying a starting and ending frame to use, and skipping any empty frames that may be at the start of the Groom animation.

Using a Groom Cache in Unreal Engine

After you import a Groom Cache, it can be applied to a **Groom Cache** assignment slot of a **Groom Component**.
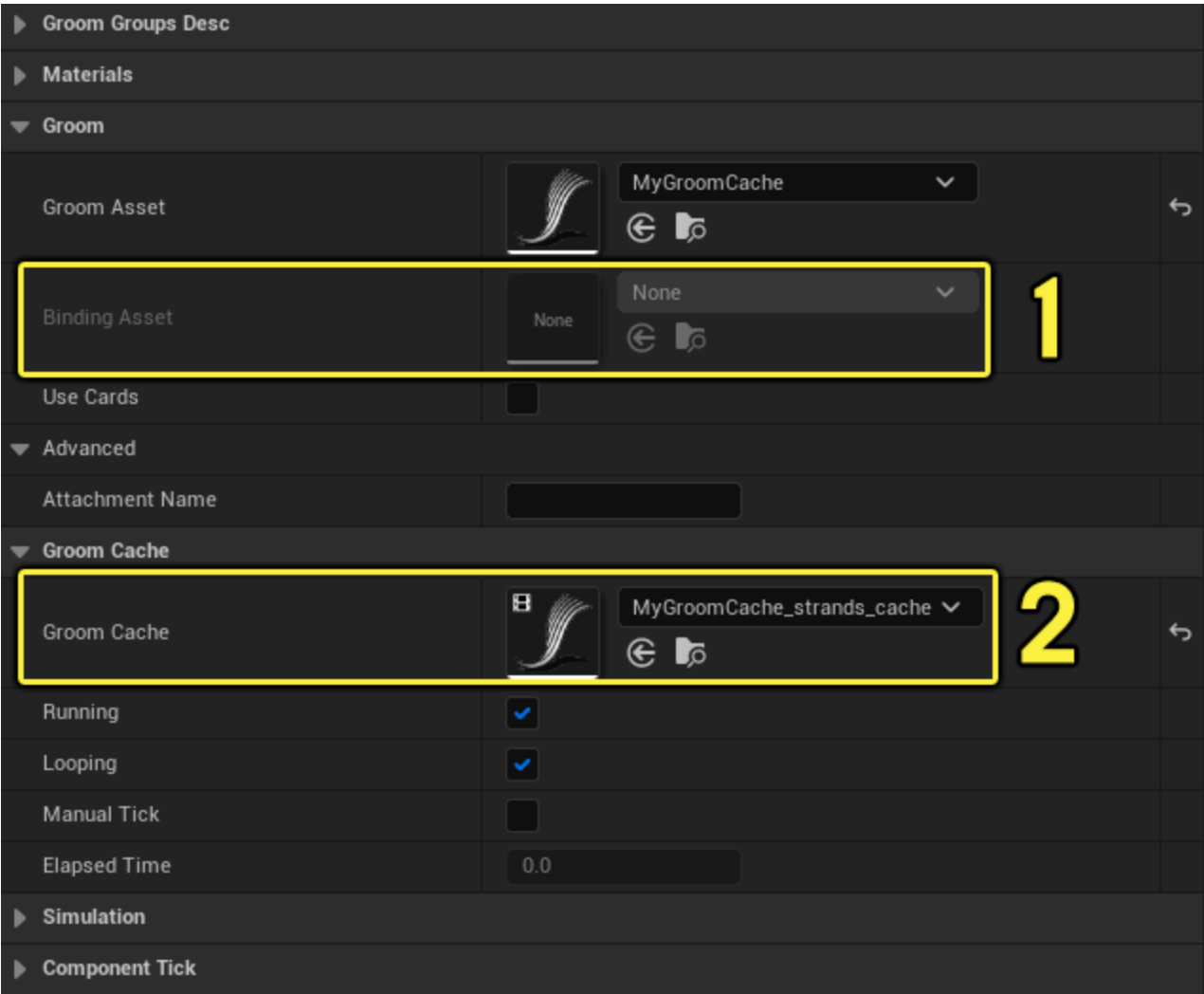

> ⓘ Groom Caches must be used in conjunction with a topologically compatible Groom Asset, since they do not contain the render data or resources to render the strands of the groom.

You can't assign a **Groom Cache** (2) to a Groom Component if it already has a **Groom Binding** (1) assigned, the Groom Cache assignment slot will be grayed out.

> ⓘ For Guides Groom Caches, the Groom Asset must enable simulation for each **Groom Group** for the Groom Animation to be visualized while using Play-in-Editor (PIE).

Using a Groom Cache in a Level Sequence

A Groom Cache can be used and controlled from a [Level Sequence](#) by adding a **Groom Cache** track on the Actor with a Groom Component.
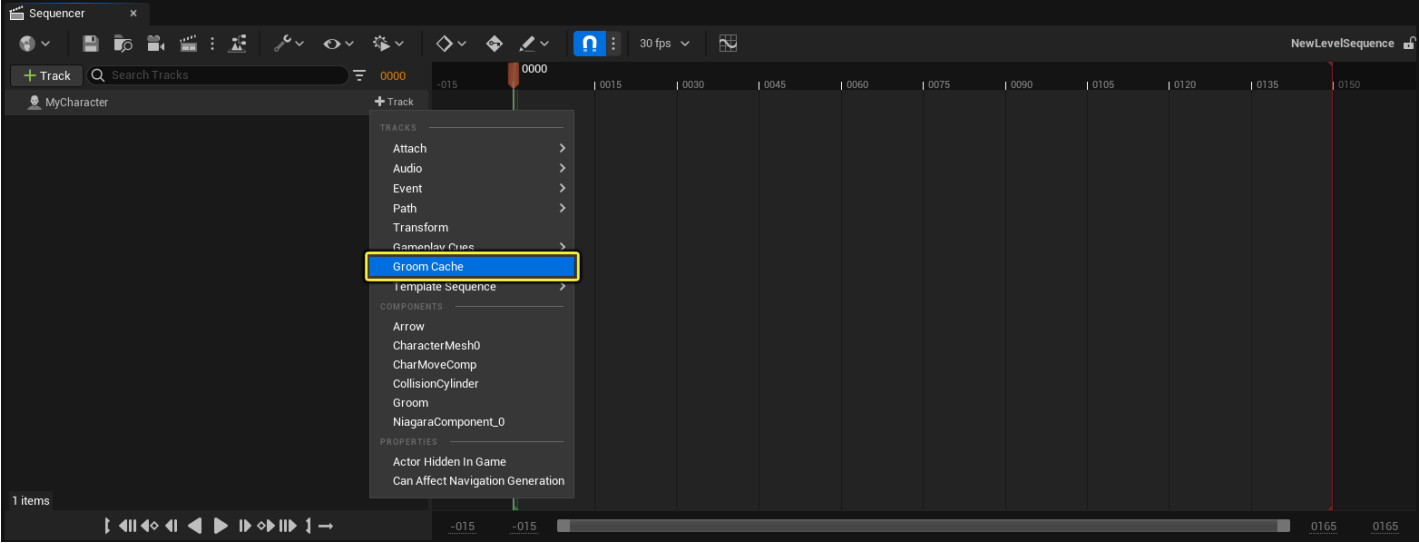
If the Groom Component already has a Groom Cache applied to it, the Groom Cache section on the track automatically sizes to fit the animation range. Otherwise, you have to size it manually.

Sequencer | NewLevelSequence

MyCharacter

Groom Cache

MyGroomCache_strands_cache

0000   0015   0030   0045   0060   0075   0090   0105   0120   0135   0150

+ Track   Search Tracks   0000   30 fps