

# Alembic for Grooms Specification

Describes guidelines for exporting Grooms as Alembic files for use with Unreal Engine.



The Hair Rendering and Simulation features are intended to provide a minimally viable implementation of strand-based hair importing, rendering, and simulation. This update does not provide a hair grooming solution. Hair shaping, or grooming, will need to be performed in an external DCC tool such as Maya by using their XGen tool or the plugin [Shave and a Haircut](#). Other tools like [Ornatrrix](#), [Yeti](#), or [Houdini](#) can be used as well.

This document outlines an informal schema for Alembic files in an attempt to standardize the import of hair groom and guide hair information into Unreal Engine for use with the [Hair Rendering and Simulation](#) features. The intent is that as you generate Hair Grooms, conforming to this schema allows Unreal Engine to import those grooms directly. Upon import, attributes that are supported by UE4 for both rendering and physics will be mapped to those attributes in the Alembic file.

## Goals

1. Define a groom-specific, informal, Alembic metadata schema to facilitate importing of hair into Unreal Engine.
2. Examine the attributes that are supported by Unreal Engine for hair, and map those attributes onto Alembic.

## Curves

The general shape of the hair is read from the standard Alembic Curve schema:

```
Alembic::AbcGeom::ICurves
```



Copy full snippet

# Width



Groom attributes exported using this specification enables the importer to retrieve and use them when building the groom. However, the width attribute does have a special behavior. For example, in a DCC application like Maya, width values are exported directly on the curves so there's no need to export a custom `groom_width` attribute; the importer will convert Maya's width values into the `groom_width` attribute. It is worth noting that if the `groom_width` attribute is present in your groom, it will not be overwritten during import. If no `groom_width` attribute is specified, or one couldn't be converted from width values, the builder in Unreal Engine falls back to using a value of 1 centimeter for the width value.

The curve width can be stored per-ICurve using:

```
(GeometryScope::kConstantScope)
```

Copy full snippet

Or it can be stored per-curve using:

```
(GeometryScope::kUniformScope)
```

Copy full snippet

Or it can be stored per-cv using:

```
(GeometryScope::kVertexScope)
```

Copy full snippet



Note that the importer supports `kUniformScope`, but this has not been tested.

# Naming Conventions

Property names must be lowercase with no spaces or special characters.

All properties must start with the following prefix:

```
groom_<attr>
```

Copy full snippet

For example, `groom_id`, `groom_color`, or `groom_width` are all using the proper naming conventions.

# Scoping and Precedence

Properties in this document have an assigned scope. The scope controls at which *level* the groom property can be defined.

The available scopes are:

Scope	Description
<b>Constant</b>	These parameters have a single value for all curves with an <code>AbcGeom::ICurves</code> Object.
<b>Uniform</b>	These parameters have a value per-curve in an <code>AbcGeom::ICurves</code> Object.
<b>Vertex</b>	These parameters have a value per-curve per-cv.

## General Properties

This section outlines properties that apply to the groom as a whole. These properties should be added to the topmost node: IXForm or ICurves.

Name	Type	Optional	Description	Value
<code>groom_version_major</code>	int16	No	Used to identify which <b>major</b> version of the groom schema this file conforms to.	1
<code>groom_version_minor</code>	int16	No	Used to Identify which <b>minor</b> version of the schema this file conforms to.	5
<code>groom_tool</code>	string	Yes	The name and version of the tool that generated this file. Useful for debugging and tracking down issues.	
<code>groom_properties</code>	string	Yes	String describing the parameters used to generate this groom. These are tool-specific options, used for debugging.	

## Geometry Parameters



Name	Type	Scope	Optional	Description	Range
<div><div>groom_close</div><div>st_guides</div></div>	int32[3]	Uniform	Yes	Intended for specialized use where the interpolation data is computed outside of Unreal Engine.	[0, INT_MAX]
<div><div>groom_guide</div><div>_weights</div></div>	float32[3]	Uniform	Yes	Intended for specialized use where the interpolation data is computed outside of Unreal Engine.	[0, FLOAT_MAX]