

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Animating Characters and Objects

/ Skeletal Mesh Animation System

/ Animation Debugging and Optimization

/ Animation Compression

/ Animation Compression Library

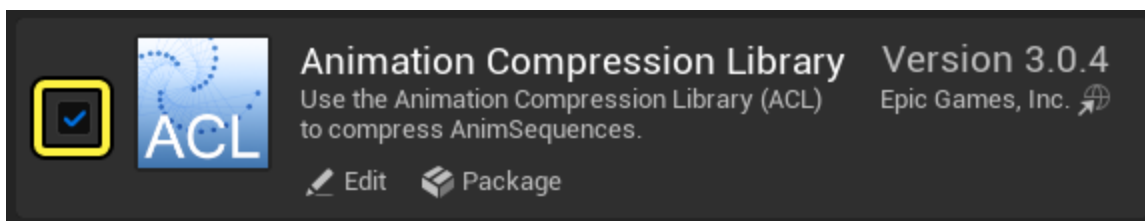
# Animation Compression Library

Overview of the Animation Compression Library plugin.

The **Animation Compression Library (ACL)** is a plugin that introduces more robust and customizable compression codecs that you can use to further customize your animation compression.

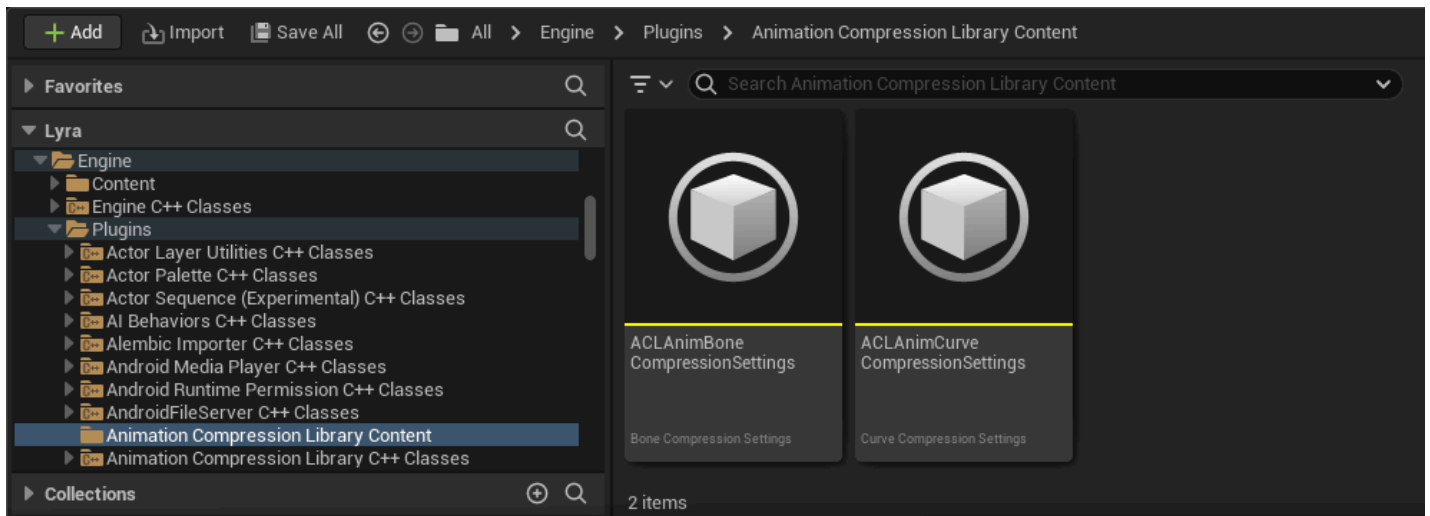
## Prerequisites

- Ensure the **Animation Compression Library** plugin. Navigate in the Menu Bar to **Edit > Plugins** and locate the Animation Compression Library, listed under the Animation section, or by using the Search Bar. Enable the Plugin and restart the editor.

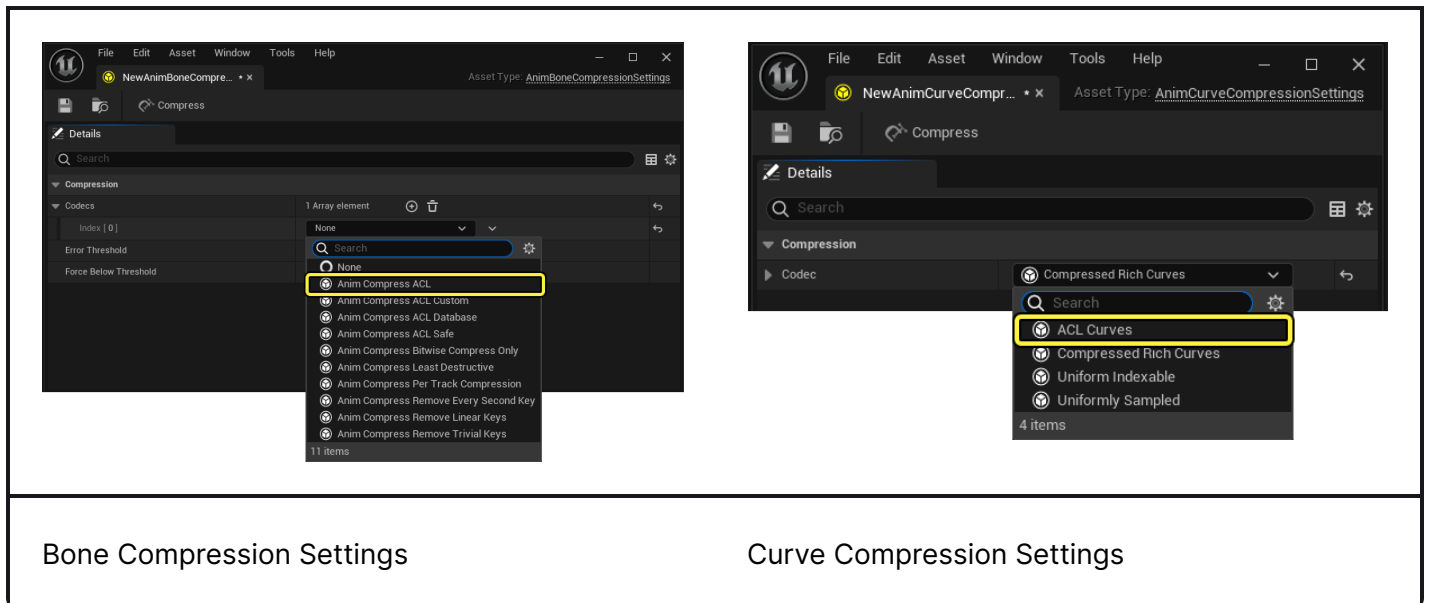


## ACL Set Up

The ACL Plugin comes pre configured with both a Bone and Curve Compression Settings assets that you can use to compress any of your project's Animation Sequences. You can access these Compression Settings assets in the following location: [Engine > Plugins > Animation Compression Library Content](#).



Additionally you can create new Bone and Curve Compression Settings assets and set the **Codec** property to use the **Anim Compress ACL** option to access its functionality



It is recommended to use the default **Anim Compress ACL** codec for most projects, which is designed to produce the most optimal results for most use cases.

## Changing the Default Compression Settings

To change the default Compression Settings asset that is used during packaging, you will need to edit the `BaseEngine.ini`, or equivalent file, in your project under the **Animation.DefaultObjectSettings** section.

The relevant entry is the:

```
BoneCompressionSettings="/Engine/Animation/DefaultAnimBoneCompressionSettings".
```

This entry points to the default Compression Settings assets contained within the **Engine Content** folder. You can change it to point to any other suitable asset of your choice by modifying the entry to use a different path, such as one that points to the default ACL Compression Settings assets, or any custom Compression Settings assets.

To set the default compression settings asset to use the default ACL Compression Settings asset, use the following entry path:

```
BoneCompressionSettings="/Plugins/AnimationCompressionLibraryContent/ACLAnimBoneCompressionSettings"
```



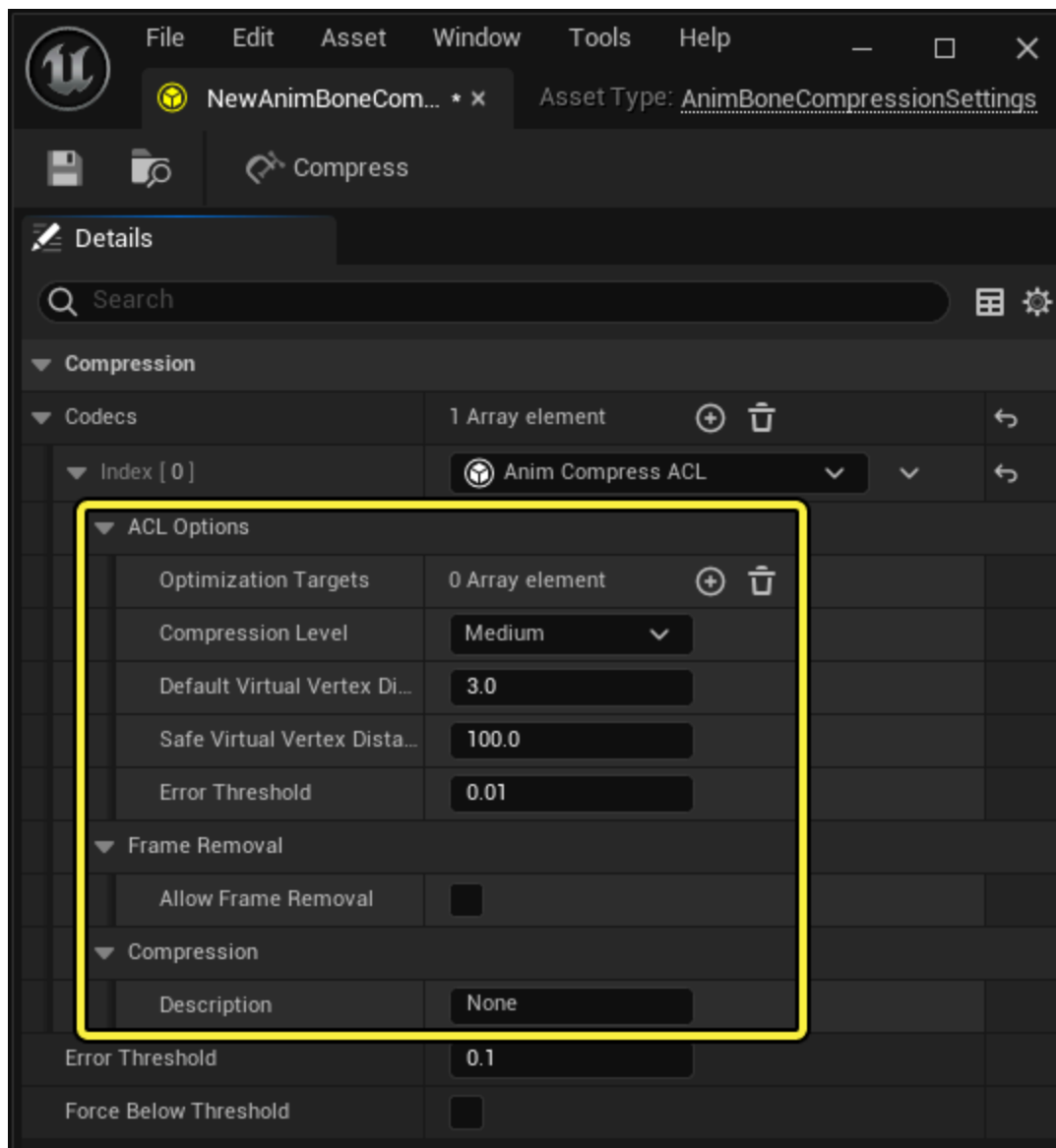
To set the entry to use a custom asset, the path will point to wherever your asset is stored. For example, `BoneCompressionSettings="/Game/Compression/MySettings"` for a file located in the following file path: `.../MyProject/Content/Compression/MySettings.uasset`.

## ACL Bone Compression

The Anim Compress ACL codec favors as safe a compression as possible and so it is not recommended to perform much settings tuning. If you need more powerful compression or wish to explore more robust compression features, you can optionally use the **Anim Compress Custom ACL** codec which allows for more advanced options and settings tuning and is designed to be used for debugging purposes.

## ACL Bone Compression Reference

Here you can reference a list of the ACL Codec properties and a description of their function. All units are in Unreal Units (cm).



## Property

## Description

### Optimization Targets

The Skeletal Mesh used to estimate the skinning deformations during compression. To add an asset, use (+) **Add** and then select an asset from the drop-down menu. This property is meant to ensure optimal visual fidelity is retained during compression. If you can provide The ACL codec with a Skeletal Mesh reference, the codec will not have to approximate the visual mesh, improving the results of compression. However, with higher fidelity meshes, the memory footprint tends to increase slightly.

### Compression Level

Set the Compression level to use from the five available levels. Higher levels will be slower to compress but can yield a lower memory footprint. It is important to test your compression results to find the best results for your project.

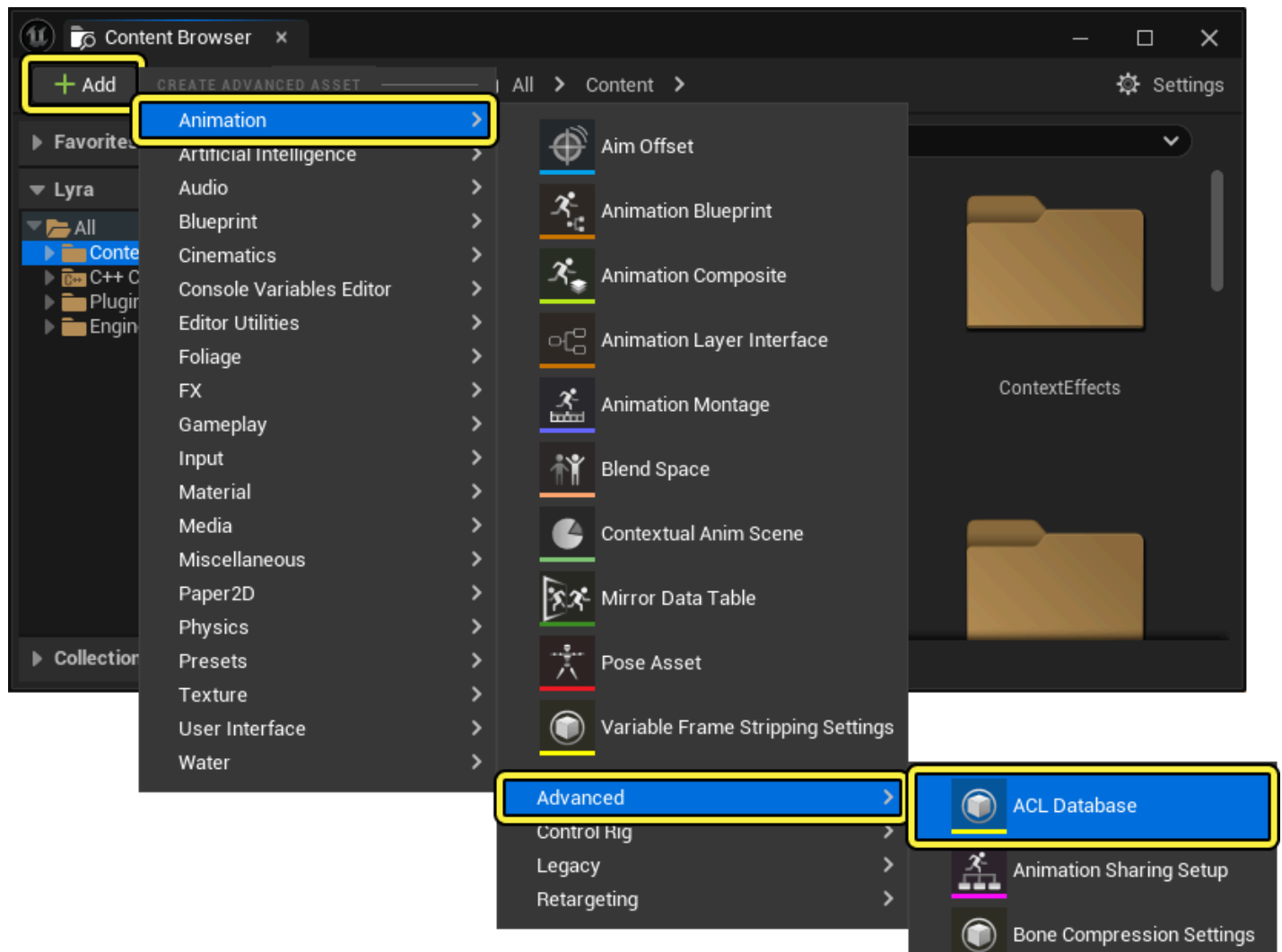
Property	Description
	<p>The compression level dictates how aggressively ACL tries to optimize the memory footprint. Higher levels will yield a smaller memory footprint but take longer to compress while lower levels will compress faster with a larger memory footprint. Medium strikes a good balance and is suitable for production use.</p>
<b>Default Virtual Vertex Distance</b>	<p>Set the Default virtual vertex distance for normal bones.</p> <p>Choosing an appropriate value for the Default Virtual Vertex Distance is important. The default value of 3cm is generally suitable for ordinary characters but large objects or exotic characters might require fine tuning. UE also has support for special bones that require more accuracy. By default, every bone that has a socket attached will be deemed as needing high accuracy as well as any bone that contains one of the substrings present in <code>UAnimationSettings::KeyEndEffectorsMatchNameArray</code>. Common substrings included are: hand, eye, IK, camera, etc. For those special bones, the Safe Virtual Vertex Distance is used instead.</p>
<b>Safe Virtual Vertex Distance</b>	<p>The virtual vertex distance for bones that require extra accuracy.</p>
<b>Error Threshold</b>	<p>The error threshold to use when optimizing and compressing the animations sequence. Lower values allow for less error, while higher values allow for less error.</p> <p>The ACL optimization algorithm will attempt to aggressively remove everything it can until the error exceeds a specified Error Threshold. For this reason, the threshold is very important and it should be very conservative. A default value of 0.01cm is appropriate for cinematographic quality and most likely does not require any tuning. The error threshold works in conjunction with the virtual vertex distance since the error is measured on the virtual vertices.</p>

# Anim Compress ACL Database

The Anim Compress ACL Database codec exposes quality based streaming to the Compression Settings asset that can be used to selectively strip the least important keyframes of your animations that can then be optionally streamed back in or out at runtime, effectively controlling quality through streaming.

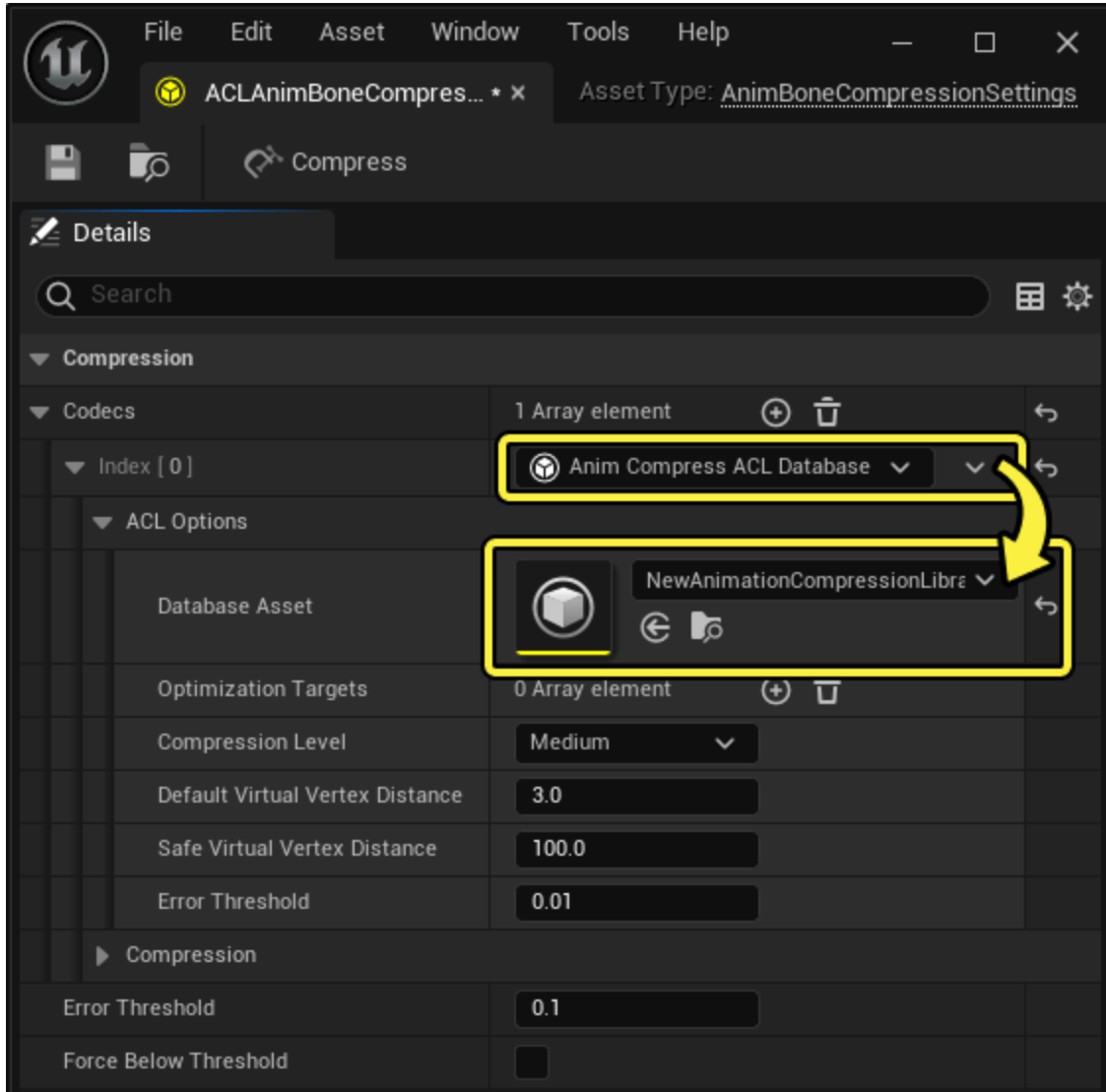
## ACL Database Setup

To use the ACL Database codec, you must first create a new **ACL Database** asset, using (+) **Add** in the Content Browser and navigating to **Animation** → **ACL Database**.

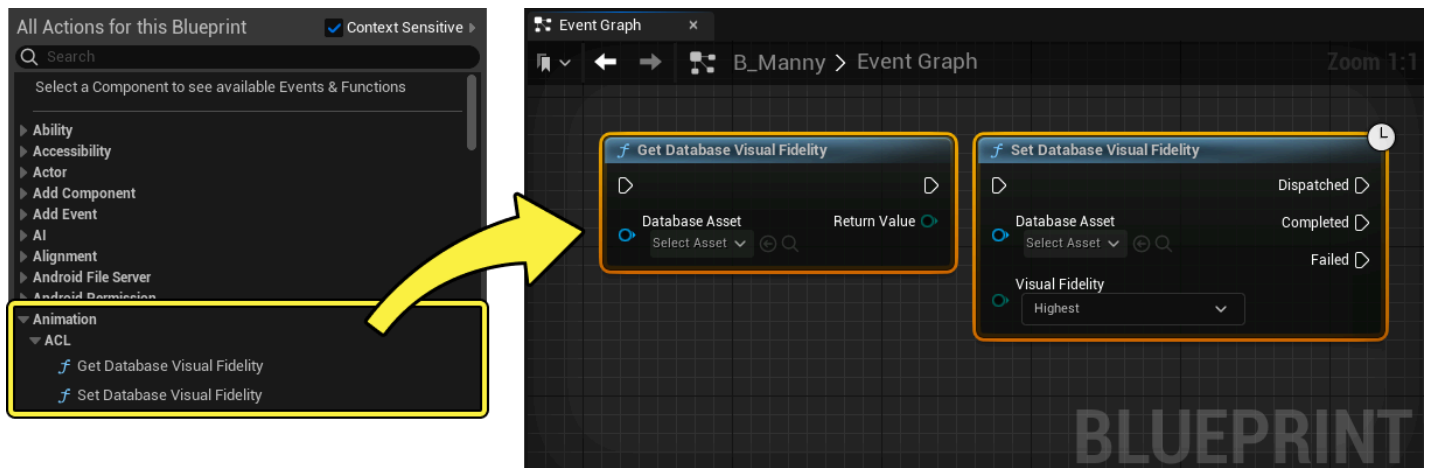


After creating the asset you can double-click the asset to expand its settings. For more information see the [ACL Database Settings](#) section.

Once your database asset is configured, it must be referenced in the Anim Compress ACL Database codec's Database Asset property.



The Database Asset can then be controlled using Blueprints, to dynamically set the ACL Custom codec's **Compression Level**. You can add ACL Database nodes to Blueprints by right-clicking the graph and navigating in the context menu to **Animation > ACL**.



## Get Database Visual Fidelity

You can use the Get Database Visual Fidelity node to read what the defined ACL Database asset's Visual Fidelity is currently set to. To define a Database Asset, you can either connect a reference variable using the **Database Asset** input pin, or use the node's drop-down menu to select a ACL Database asset. The read visual fidelity is returned as an enum value from the **Return Value** output pin

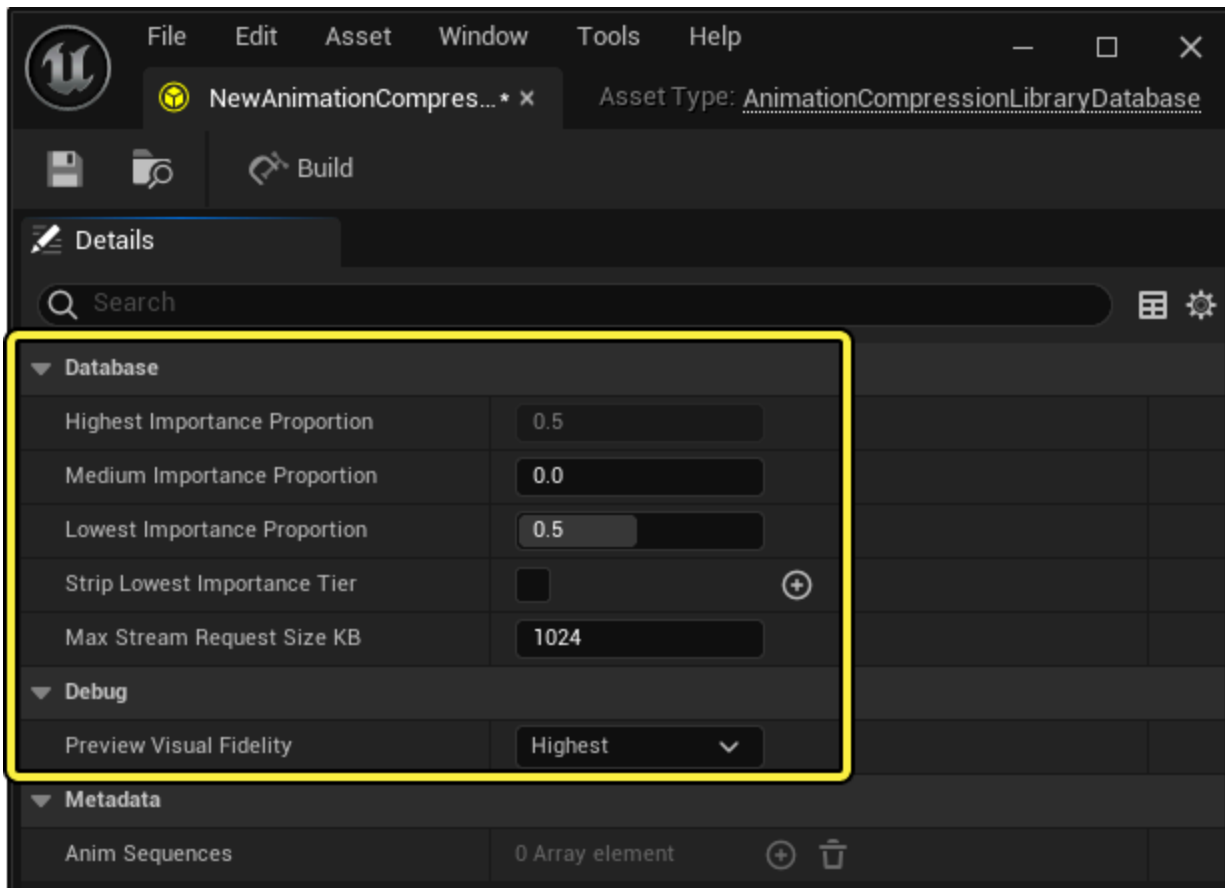
## Set Database Visual Fidelity

You can use the Set Database Visual Fidelity node to set the ACL Database asset's visual fidelity using Blueprints. To define a Database Asset, you can either connect a reference variable using the **Database Asset** input pin, or use the node's drop-down menu to select a ACL Database asset. To set the visual fidelity, connect a reference variable using the **Visual Fidelity** input pin, or define the desired level using the drop-down menu on the pin in the graph.

## ACL Database Settings

An ACL Database asset is used to reference all the Animation Sequence assets for the codec automatically. Here you can reference a list of the ACL Database assets settings and a description of their function:





Setting	Description
<b>Highest Importance Proportion</b>	Percentage of animation data that remains always loaded in memory (the most important key frames). All three proportion fields must sum to 1.0.
<b>Medium Importance Proportion</b>	Percentage of animation data that is moved into the streamable medium tier. All three proportion fields must sum to 1.0.
<b>*Low Importance Proportion*</b>	Percentage of animation data that is moved into the streamable lowest tier. All three proportion fields must sum to 1.0.
<b>Strip Lowest Importance Tier</b>	Whether or not to strip entirely the lowest tier (once stripped, it cannot be streamed).
<b>Max Stream Request Size KB</b>	Maximum IO stream request size (small reads perform more poorly and should be avoided).

Setting	Description
<b>Preview Visual Fidelity</b>	Which visual fidelity level to use for preview in the editor (editor only, transient). The preview visual fidelity field is meant to help preview in the editor what the animation quality will be once data is streamed at a particular fidelity level. By default, the editor always shows the highest visual fidelity.

## ACL Database Codec Settings

This codec is identical to the Anim Compress ACL codec with the added ability to reference an ACL Database asset which you can use to reference multiple Animation Sequences. Multiple ACL Database codecs can reference the same database assets. Animation Sequences that use this codec will end up in the selected database and can have their data streamed at runtime or stripped entirely during cooking.

Frame stripping with ACL is much more powerful than the UE frame stripping. The ACL allows you to control how much data you want to strip and it will pick the least important keyframes from all animations within the database. This means that some sequences might retain more keyframes than others if they are more important which could mean that ACL framestripping can result in far less destructive compression and can also be globally optimized across many sequences.



It is recommended that you test your frame stripping results when compressing animations. While the ACL will strip keyframes that contribute the least error, it may not perform as well for individual sequences when compared to compressing entire animation sets. The error may be lower, but the perceived fidelity loss may be higher than using Unreal Engines default frame stripping techniques. Unreal Engine's frame stripping targets a fixed frame rate, which removes keyframes in a uniform manner. While this method does not attempt any preservation of more important key frames, resulting in higher error, the perceived fidelity loss could be lower than the ACL solution.

In a cooked build, no data will be streamed by default. The default visual fidelity level is at its lowest. In order to increase it, data must be streamed. This is exposed through the blueprint interface.

# ACL Blueprint Streaming

The visual fidelity can be queried and set through ordinary latent blueprint nodes. By setting the desired fidelity level, ACL figures out what needs to be streamed in or out. If multiple change requests come in while streaming is in progress, they will be queued and executed once everything else queued prior has completed. It is not currently possible to interrupt a fidelity change request.

When the visual fidelity changes, memory is allocated and freed on demand to accommodate the request. Data is loaded from disk asynchronously.

## Anim Compress Custom ACL

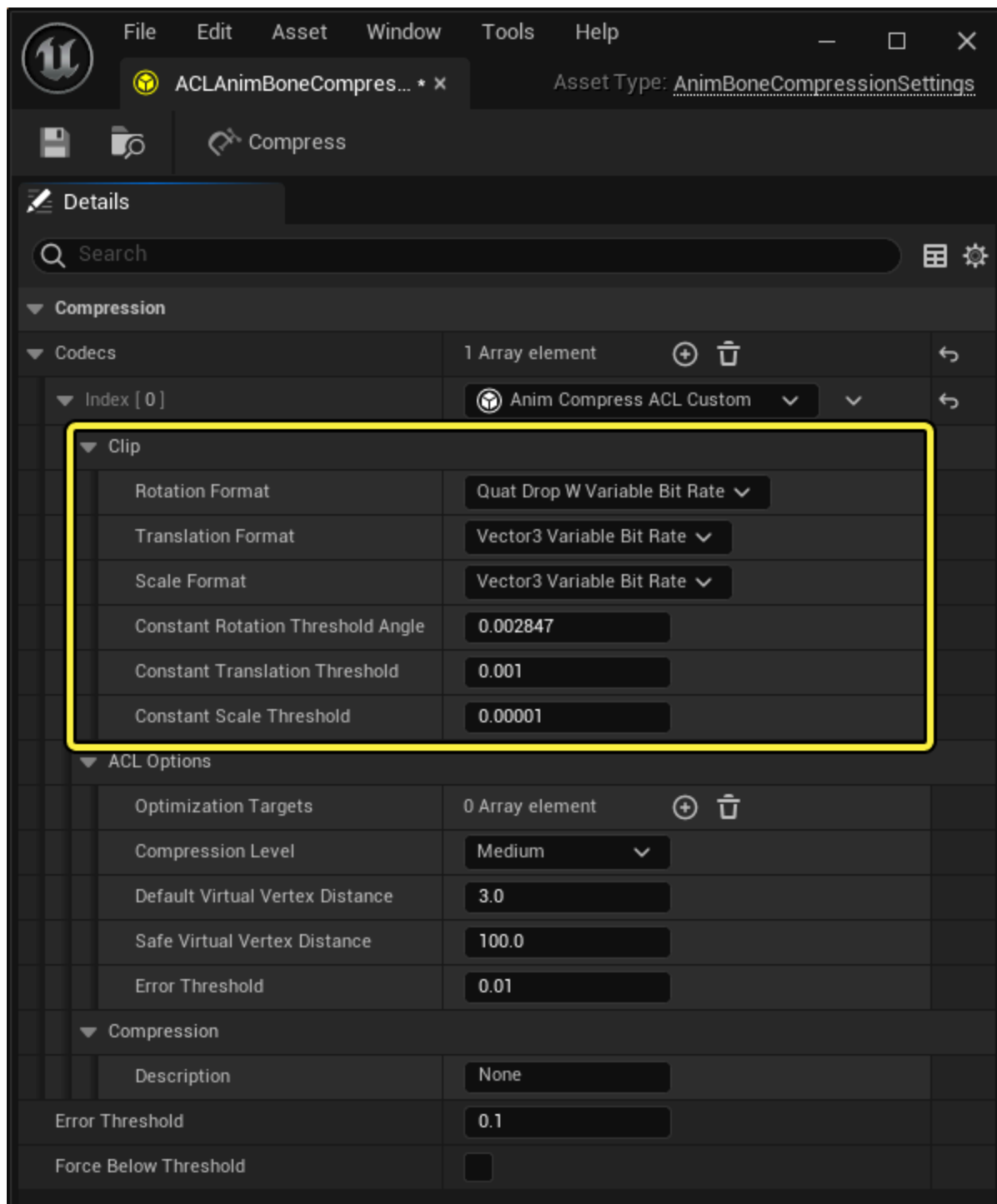
Using the custom ACL compression codec allows you to tweak and control every aspect of the ACL.



The Custom ACL codec is provided mostly for debugging purposes and therefore should not be needed in production. It's also important to note, that as a result of supporting every option possible, decompression can often end up being a bit slower as less code is stripped by the compiler.

## Custom ACL Settings

Here you can reference a list of the Anim Compress Custom ACL codec's properties and a description of their functionality.



Property

Description

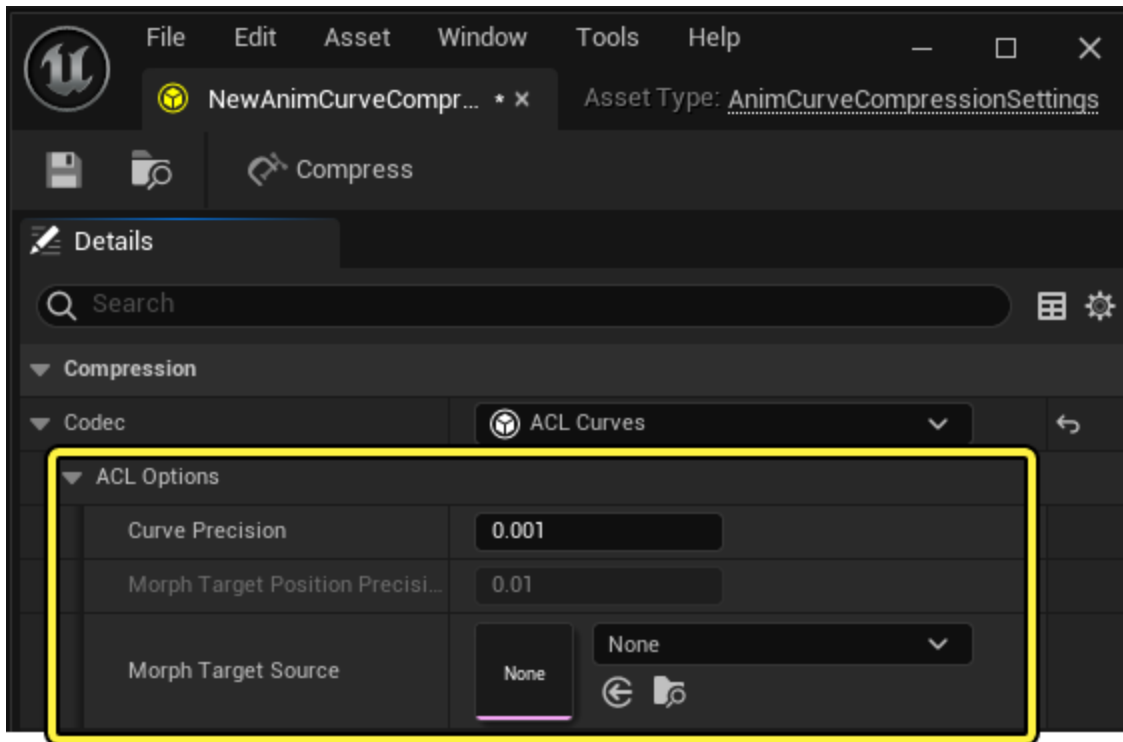
### Rotation Format

Select the rotation format to use. The ACL plugin supports three rotation formats: Quat Full Bit Rate, Quat Drop **W** Full Bit Rate, and Quat Drop **W** Variable Bit Rate. The Quat Drop **W** Variable Bit Rate option is almost always the best choice which is why they are set as the defaults. The Quat Full Bit Rate, and Quat Drop **W** Full Bit Rate options can be used as a safety fallback and for debugging purposes. You can

Property	Description
	<p>select a format from the following options, using the drop down-menu:</p> <ul style="list-style-type: none"> <li>• <b>Quat Full Bit Rate:</b> Compressed data to 32 bits floats, which essentially retains raw animation data.</li> <li>• <b>Quat Drop W Full Bit Rate:</b> Compresses data to <b>X</b>, <b>Y</b>, and <b>Z</b> 32 bit floats and drops <b>quaternion (W)</b>.</li> <li>• <b>Quat Drop W Variable Bit Rate:</b> Compresses data using a variable bit rate to store <b>X</b>, <b>Y</b>, and <b>Z</b> data, dropping the <b>quaternion (W)</b>. The variable bit rate is the ideal choice and the other option is provided for debugging purposes.</li> </ul>
<b>Translation Format</b>	<p>Select the translation format to use. You can select a format from the following options, using the drop down-menu:</p> <ul style="list-style-type: none"> <li>• <b>Vector3 Full Bit Rate:</b> A debug option that uses the full bit rate of the Vector 3 value.</li> <li>• <b>Vector3 Variable Bit Rate:</b> The variable bit rate is the ideal choice and the other option is provided for debugging purposes.</li> </ul>
<b>Scale Format:</b>	<p>Select the scale format to use. You can select a format from the following options, using the drop down-menu:</p> <ul style="list-style-type: none"> <li>• <b>Vector3 Full Bit Rate:</b> A debug option that uses the full bit rate of the Vector 3 value.</li> <li>• <b>Vector3 Variable Bit Rate:</b> The variable bit rate is the ideal choice and the other option is provided for debugging purposes.</li> </ul>
<b>Constant Rotation Threshold Angle:</b>	Set the threshold used to detect constant rotation tracks.
<b>Constant Translation Threshold:</b>	Set the threshold used to detect constant translation tracks.
<b>Constant Scale Threshold:</b>	Set the threshold used to detect constant scale tracks.

# ACL Curve Compression Reference

You can use the ACL Curve compression codec to compress animation curve data for things like [Animation Modifiers](#) and [Morph Targets](#). The ACL Curve codec works especially well when compressing Morph Targets if you provide a Skeletal Mesh as a reference for the compression codec. With a reference mesh, the codec can calculate and store a vertex displacement precision value instead of a generic scalar track precision value to compress mesh deformation curve tracks. Here you can reference a list of the ACL Curve Compression Codec properties and a description of their function.



Property	Description
<b>Curve Precision</b>	The curve precision to target when compressing the animation curves, if there is not an animation curve associated with the present Morph Targets in the source asset.
<b>Morph Target Position Precision</b>	When the <b>Morph Target Source</b> property has a definition, set the desired precision of morph target curves in world space units to target when compressing morph target animation curves. This guarantees that morph target deformations meet the specified precision value (0.01 cm is the default).

Property

Description

**Morph Target Source**

Set a Skeletal Mesh that is used to extract the Morph Target curves to be compressed using the **Morph Target Position Precision** value. If an Animation Curve is mapped to a Morph Target within the source asset, the **Morph Target Position Precision** property's value will be used to calculate the compression precision. If there is not a Animation Curve present, then the **Curve Precision** will be used instead.

To add a Skeletal Mesh asset, use the drop-down menu, and select the applicable asset.