

From the [Alembic home page](#):

Alembic is an open computer graphics interchange framework. Alembic distills complex, animated scenes into a non-procedural, application-independent set of baked geometric results. This ‘distillation’ of scenes into baked geometry is exactly analogous to the distillation of lighting and rendering scenes into rendered image data.

Alembic is focused on efficiently storing the computed results of complex procedural geometric constructions. It is very specifically **not** concerned with storing the complex dependency graph of procedural tools used to create the computed results. For example, Alembic will efficiently store the animated vertex positions and animated transforms that result from an arbitrarily complex animation and simulation process which could involve enveloping, corrective shapes, volume-preserving simulations, cloth and flesh simulations, and so on. Alembic will not attempt to store a representation of the network of computations (rigs, basically) which are required to produce the final, animated vertex positions and animated transforms.

In brief, Alembic can be used to write an animated mesh to a drive, and read it back quickly and efficiently. This means that a mesh can be animated with very CPU-intensive rig and then ‘baked’ to an Alembic file. Finally it can be loaded into the shot file for shading and lighting with only moderate CPU usage.

Due to the Open Source nature of the Alembic standard as well as the C++ library implementing that standard, **Blender can be used in a hybrid pipeline**. For example, other software, such as Houdini or Maya, can export files to Alembic, which can then be loaded, shaded, and rendered in Blender. It is also possible to animate characters (or other models) in Blender, export to Alembic, and load those files into other software for further processing.

Importing Alembic Files

When importing an Alembic file, [Mesh Sequence Cache modifiers](#) are automatically added to time-varying meshes. For time-varying object transforms (animation of rotation, location, or scale) the [Transform Cache Constraint](#) is used.

General

Scale

Value by which to enlarge or shrink the objects with respect to the world’s origin

Options

Relative Path

Select the file relative to the blend-file.

Set Frame Range

If checked, update scene’s start and end frame to match those of the Alembic archive.

Is Sequence

Set to true if the cache is split into separate files.

Validate Meshes

Check the imported mesh for corrupt data and fix it if necessary. When disabled, erroneous data may cause crashes displaying or editing the meshes. This option will make the importing slower but is recommended, as data errors are not always obvious.

Always Add Cache Reader

Add cache modifiers and constraints to imported objects even if they are not animated so that they can be updated when reloading the Alembic archive.

Exporting to Alembic Files

This section describes the effect of the different export options.

General

Scale

This sets the global scale of the Alembic file. Keep it at the default value of 1.0 to use Blender's units.

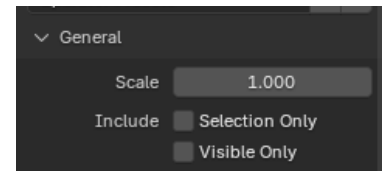
Include

Selected Objects

When enabled, exports only the selected objects. When disabled, all objects are exported.

Visible Objects

Limits the export to scene collections that are currently visible.



General options.

Scene

Frame Start, End

Sets the frame range to export to Alembic. This defaults to the current scene frame range.

Sub-frame Sampling

These options control the sub-frame sampling of animations.

Samples Transform

Transform Samples sets the number of times per frame at which animated transformations are sampled and written to Alembic.

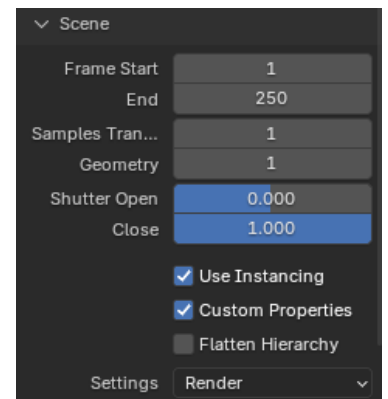
Geometry

Geometry Samples sets the same, but then for animated geometry.

Shutter Open, Close

Shutter Open/Close define the interval [open, close] over which those samples are taken. The valid range is -1 to 1, where -1 indicates the previous frame, 0 indicates the current frame, and 1 indicates the next frame.

For example, if information for detailed mesh motion blur is desired, some subframes around the current frame can be written to Alembic by using a sample count of 5, Shutter Open at -0.25 and Shutter Close at 0.25. This mimics a "180 degree" shutter, opening 90 degrees before the frame and closing 90 degrees after the frame.



Scene options.

Use Instancing

Exports data of [duplicated](#) or [instanced](#) objects as Alembic instances; speeds up the export and can be disabled for compatibility with other software.

Custom Properties

When enabled (which it is by default), custom properties are exported to Alembic as well. The following custom property types are supported:

- Numbers (`int`, `float`) and strings. These are exported as arrays of a single element, so `47` will be exported as `[47]` to Alembic, and `"Agent"` to `["Agent"]`. This matches the behavior of many other DCCs.
- Lists of numbers and strings. These are exported as-is, so `[327, 47]` is exported as `[327, 47]`.
- Matrices and nested arrays of numbers. These are flattened into one long list, so a 3×2 matrix of numbers will become a list of 6 numbers. Similarly, nested lists `[[1, 2, 3], [4, 5], [6]]` will be exported as `[1, 2, 3, 4, 5, 6]`.
- Numbers can be animated as well.

Flatten Hierarchy

When disabled, parent/child relations between objects are exported too. Any parent object that is not exported itself, but with children that *are* exported, is replaced by an empty. When enabled, parent/child relations are not exported, and transformations are all written in world coordinates.

Settings

Determines visibility of objects, modifier settings, and other areas where there are different settings for viewport and rendering.

Render:

Use Render settings for object visibility, modifier settings, etc.

Viewport:

Use Viewport settings for object visibility, modifier settings, etc.

Geometry

UV Coordinates

When enabled, UV maps are exported. Although the Alembic standard only supports a single UV map, Blender exports all UV maps in a way that should be readable by other software.

Merge UVs

When enabled, UVs sharing the same vertex and location will be merged into a single UV. The exported file can be slightly smaller.

Normals

When enabled, an object's [Normals](#) are exported. See [Custom Split Normals of Meshes](#) below for more information.

Color Attributes

When enabled, exports Color Attributes.

Face Sets

Exports the material names per face. The material data is not exported but only material names.

Subdivisions

Apply

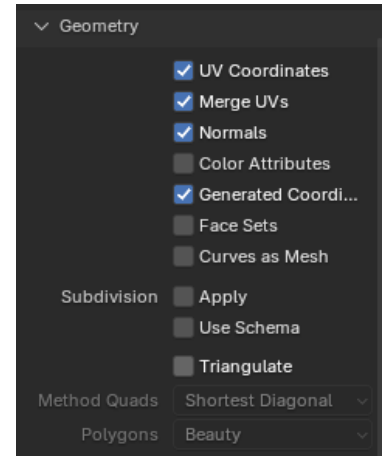
Applies any [Subdivision Surface modifiers](#) before writing to Alembic.

Use Schema

Writes polygonal meshes using the “SubD” Alembic schema, rather than the “PolyMesh” schema. This sets an import option for the program with which the file is opened, to apply its form of a non-destructive subdivision.

Triangulate

Triangulates the mesh before writing to Alembic. For more detail on the specific option see the [Triangulate modifier](#).



Geometry options.

Particle Systems

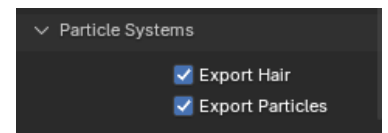
Alembic has no support for Particle Systems, in the same way that it does not support armatures.

Export Hair

Hair is exported as animated zero-width curves.

Export Particles

Particles are exported as animated points.



Particle Systems options.

Custom Split Normals of Meshes

Blender supports the import and export of [custom normals](#) to Alembic files. As a basic rule of thumb, a completely smooth mesh will be exported without normals and thus produce the smallest Alembic file. This is reflected in the importer; an Alembic mesh without normals is loaded as a smooth mesh.

On export, for every mesh:

- If it has *Custom Loop Normals* then the loop normals are exported.
- If one or more polygons are marked flat then loop normals are also exported.
- Otherwise, no normals are exported.

On import, when the Alembic mesh contains:

- Loop normals (`kFacevaryingScope`) are used as custom loop normals.
- Vertex normals (`kVertexScope` or `kVaryingScope`) are convert to loop normals, and handle as above.
- If there are no normals then the mesh is marked as smooth.
- Unsupported normal types (`kConstantScope`, `kUniformScope`, `kUnknownScope`) are handled as *no normals*.

When an imported mesh does not contain normals, the final look can be controlled using the [Normal's Shading](#).

Handling Time

Unlike Blender and many other applications and file formats, Alembic files don't have any concept of frames. Alembic works purely with time, and value that are sampled over time. For example, there is no way to distinguish 30 FPS with 2 samples per frame, and 60 FPS with 1 sample per frame. This has caused many developers to just [hard-coded 24 FPS](#) when reading Alembic files.

Blender uses the current scene frame rate to convert a frame number (in Blender) to a time in seconds (in Alembic). As a result, you can import an Alembic file that was produced at 120 FPS into a Blender scene that is 30 FPS and still not see any time stretching.

[Previous](#)
[Importing & Exporting Files](#)

[Copyright](#) © : This page is licensed under a CC-BY-SA 4.0 Int. License

[Ne](#)
[Collada \(Legacy\)](#)

Made with [Furo](#)

Last updated on 2025-05-10

[View Source](#)
[View Translation](#)
[Report issue on this page](#)