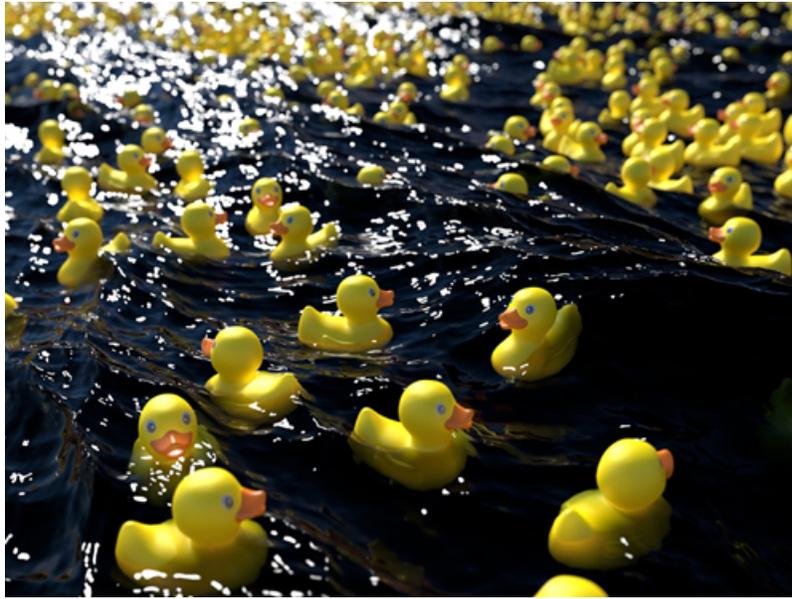


Arnold Procedural



Duck model exported as .ass file and imported as a *procedural*

You can export any object as a [.ass](#) file. A *procedural* (also called stand-in, proxy or placeholder) references the .ass file on disk (or other procedural formats supported by Arnold such as .abc, .obj, .ply, .usd etc). They allow you to keep your working scene light and workable by deferring the loading of geometry data until render time.

i During export, Arnold's current render settings will be used. Therefore, you should finalize these settings before exporting the *Stand-in*. For example, you should choose whether you wish to export with motion blur enabled or not.

w The *utility* shader's *object* mode does not work with procedurals (*stand-ins*). This is a known limitation. The color of the spheres changes when the *utility* shader's mode is *object*. However, the *stand-ins* color remains the same.

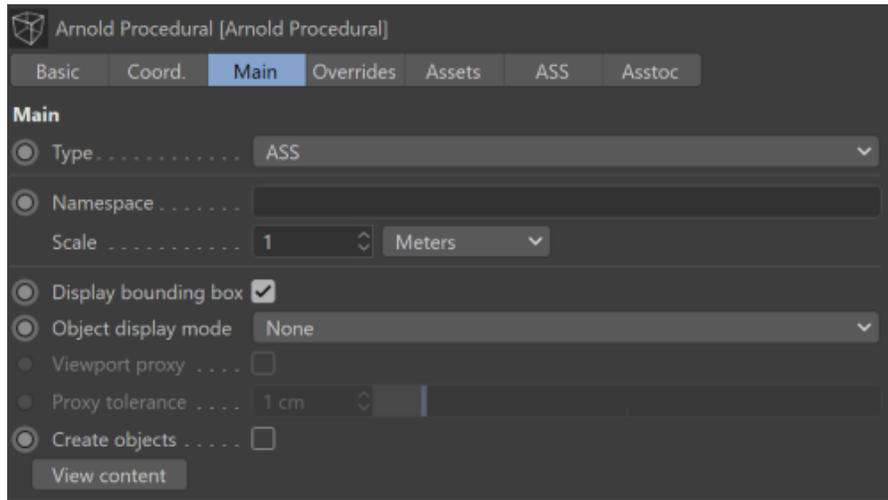


✓ Sharing Arnold Procedurals Between Plug-ins

Some users may be in a studio environment where a combination of Maya, 3DS Max, Houdini, and C4D are used. It is perfectly possible to export .ass Arnold Procedurals (stand-ins) and reuse them in other Arnold plug-ins (and vice versa), as long as Arnold can find both the plug-in and shaders. To do this, you must ensure that the Arnold shader path environment variable ARNOLD_PLUGIN_PATH is set to pick up both sets of shaders.

Arnold Procedural

To load the geometry, you need to create an **Arnold Procedural** object. Go to *Plugins > C4DtoA > Arnold Procedural* and specify the **Type** of the procedural and the archive to be used.



Procedurals are supported via the Arnold Procedural node

Type

In addition to archive **ASS** (.ass, .obj, .ply), **Alembic** (.abc) or **USD** files, pre-compiled **custom procedural plugins** can also be selected here (e.g. the **Mandelbulb** procedural), though none are shipped with C4DtoA currently.

Namespace

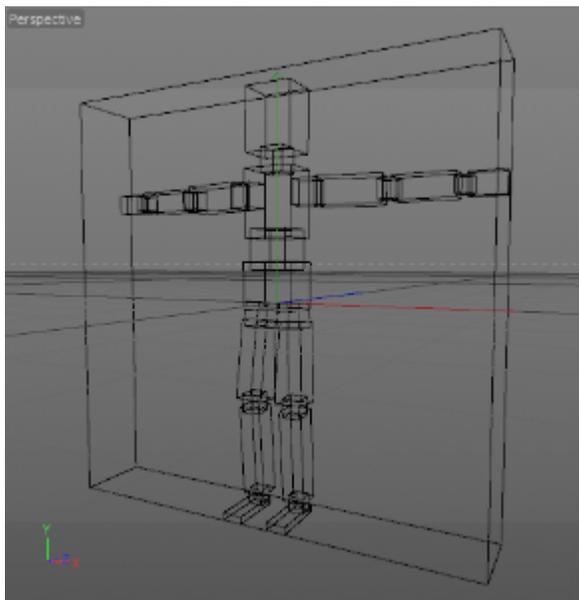
Procedurals can declare a custom namespace using this parameter. This custom namespace can be used instead of the procedural name, to reference contents through absolute or relative paths. Multiple procedurals can share the same namespace by using the same custom name. Also, they can declare an empty name and they will use the global namespace.

Scale

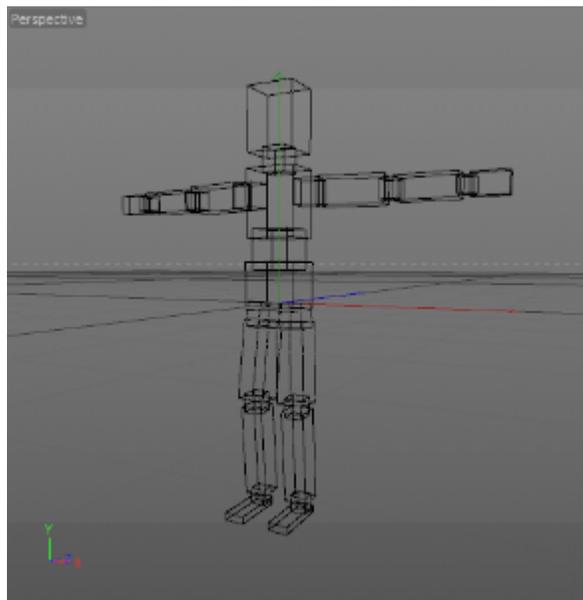
Defines the length unit of the loaded archive. Different DCCs work in different units, for instance the default unit in Maya is 1 cm, while in Houdini it's 1 m.

Display Bounding Box

Reads the bounding box from the Geometry File.



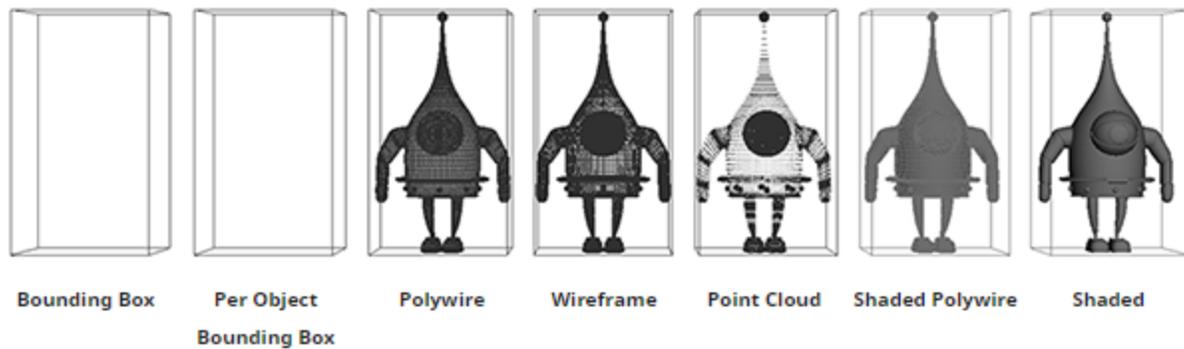
Display Bounding Box: On



Display Bounding Box: Off

Object Display Mode

The display mode of the procedural.



Viewport Proxy

Enable this option to reduce the number of points and polygons displayed in the viewport.

Proxy Tolerance

Viewport proxy objects are generated via the "Optimize Mesh" command which requires a Tolerance input. If points are closer to each other than this value, then they are merged into one point. If the points are further apart than the value, they will not be merged. If polygons become redundant (e.g., if all three corner points of the polygon occupy the same point), they will be removed.

Create Objects

If enabled the Arnold Procedural object behaves as an object generator, and geometries of the procedural will be generated to the scene as Poly objects.

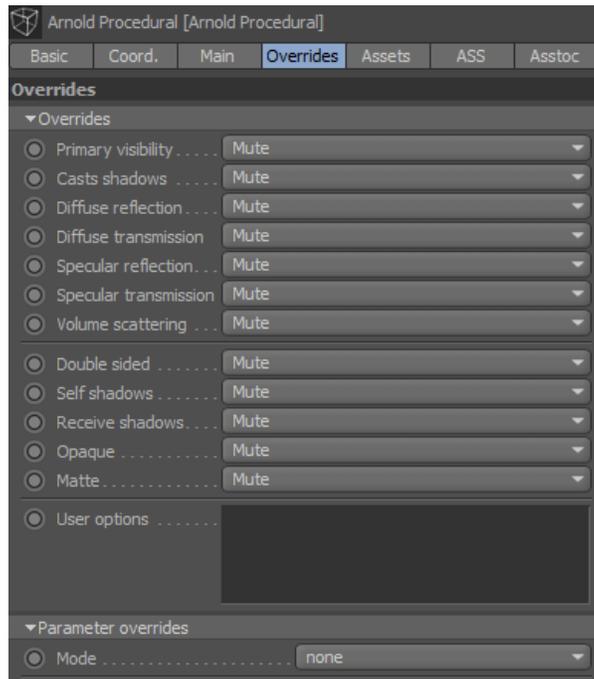
i You can use the Make object command (press c) to turn the procedural to actual polygon objects if this option is enabled.

View Content

Opens a dialog that displays the content of the procedural as a node tree. You can copy the name of an object and paste it to the selection expression of the parameter overrides (see below).

Name	Type
-/Figure	
-/Figure Figure	
-/Figure Figure Pelvis	polymesh
-/Figure Figure Pelvis Left_Thigh	polymesh
-/Figure Figure Pelvis Left_Thigh Left_Joint	polymesh
-/Figure Figure Pelvis Left_Thigh Left_Shin	polymesh
-/Figure Figure Pelvis Left_Thigh Left_Shin Left_Foot	polymesh
-/Figure Figure Pelvis Left_Thigh Left_Shin Left_Foot Left_Joint	polymesh
-/Figure Figure Pelvis Left_Thigh Left_Shin Left_Joint	polymesh
-/Figure Figure Pelvis Right_Thigh	polymesh
-/Figure Figure Pelvis Upper_Body	polymesh
-/Pyramid	polymesh
-/standard_surface	
-/standard_surface checkerboard	checkerboard
-/standard_surface standard_surface	standard_surface
default_shader	lambert

Overrides



Overrides

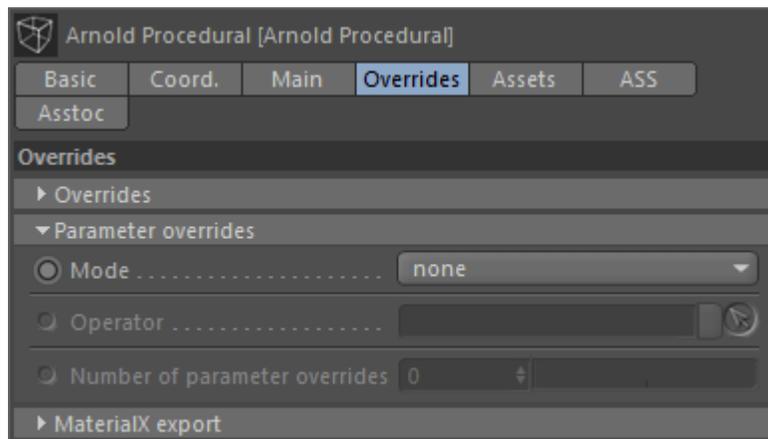
Choose to 'mute' or 'disable' various rendering options for selected Arnold Procedurals. This will override the values set within the procedural file. 'Mute' (set by default) uses the values set within the procedural file.

User Options

This string field can be set to override any parameter of the Arnold procedural node. For more info see [User Options](#).

Parameter Overrides

These options allow you to override any parameters of the Arnold nodes (shapes, shaders, lights, etc.) created by the procedural object. See this [tutorial](#) for more details.



Mode

Specify the parameter override mode, selecting from the following:

- **none**: No overrides.
- **operator network**: Use an Arnold Operator network from the object tree.
- **parameter overrides**: Specify overrides on the procedural UI.

Number of parameter overrides

Define the number of override expressions applied to the given procedural.

Enable

Enables the override.

Selection

Specify the name of the Arnold node which you want to override. Click on the button next to the input field to select a node loaded by the procedural. See the [Operators page](#) for more information on the selected expressions.



In case of an ASS procedural you can check the node names by opening the ASS file in a text editor.

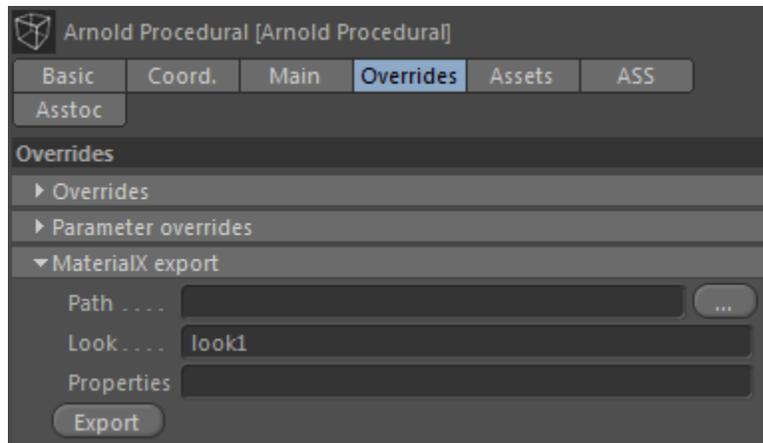
Mode

Specify the mode of the override, selecting from the following:

- **Expression:** Allows you to specify an assignment **Expression**. See the [set_parameter operator](#) page for more details.
- **Shader:** Overrides the shader of the selected shape(s) with the one specified in the **Shader** field. The **Index** field allows you to override shaders assigned to polygon selections.
- **Transform:** Overrides the transformation matrix of the selected shape(s) with the transformation of the specified object (e.g. a Null object).
- **Displacement:** Overrides the displacement shader and optionally the displacement parameters of the selected shape(s). The **Index** field allows you to override displacement shaders assigned to polygon selections.
- **Subdivision:** Overrides the subdivision **Type** and **Iterations** of the selected shape(s).
- **Color / Float / Integer parameter:** Allows you to override the value of a color / float / integer type parameter. You have to specify the name of the Arnold node parameter (**Parameter**) and the **Value**. For more information on the parameter names, see [User Options](#).

MaterialX Export

These options allow you to export the current overrides (look) to a MaterialX document (.mtlx). The export is always relative, so you can assign the look to another procedural with a different name.



Path

Choose the name and path that you want the .mtlx file to be saved as.

For existing mtlx files, the export will either update an existing look, or append a new look to the document.

Look

The look name to use in the exported MaterialX document.

Properties

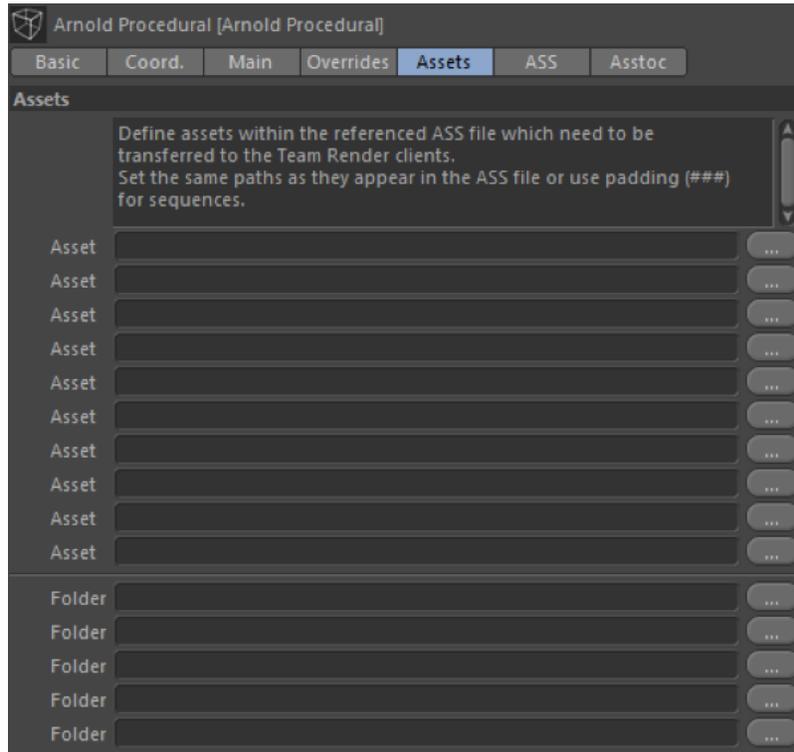
Additional shape parameters to export. For example, `step_size` and `volume_padding` for a polymesh that is rendered as a volume.

The following shape parameters are automatically exported (if they are set to non-default values):

- `visibility`
- `sidedness`
- `disp_padding`
- `disp_height`
- `disp_zero_value`
- `disp_autobump`
- `autobump_visibility`

Assets

Procedurals can reference other assets in ASS files which are 'invisible' for C4DtoA. For example, if you have an object with a texture in your referenced ASS file, C4DtoA does not know anything about this texture and won't distribute it to the clients. In this case, you have to manually define these assets on the **Assets** tab.



It is important that you define the same paths here as they are in the ASS file. If the path is absolute in the ASS file, you must set an absolute path here. If only the file name is given, you must set the file name here as well. If you have a sequence in the ASS file, define it using padding (#) characters. This is because C4DtoA has to replace the path in the ASS file with the downloaded assets on the client machines. Note, that absolute paths do not work with ass.gz files.

You can also define folders in which case all files from the folders and subfolders will be transferred to the client machines. Note that this requires relative paths in the ASS file.



Instead of defining assets here, you can keep your assets in a shared folder and add the folder to the [Texture Search Path](#) in the Render Settings.