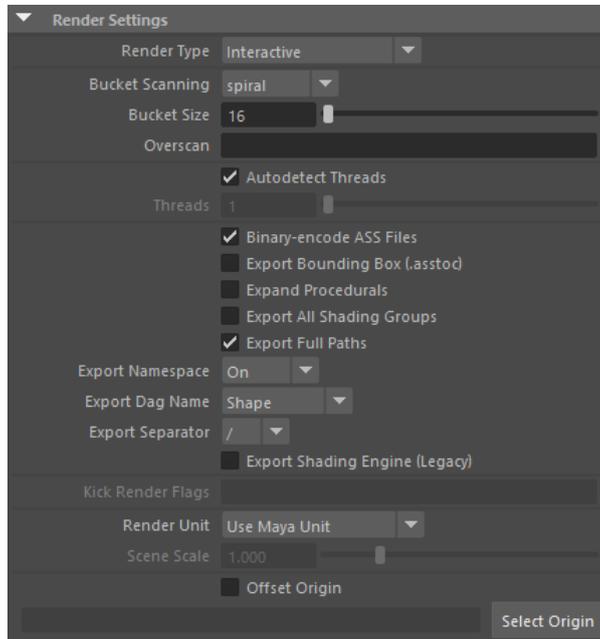


# Render Settings

The Render Settings section allows control over various Arnold **System** settings.



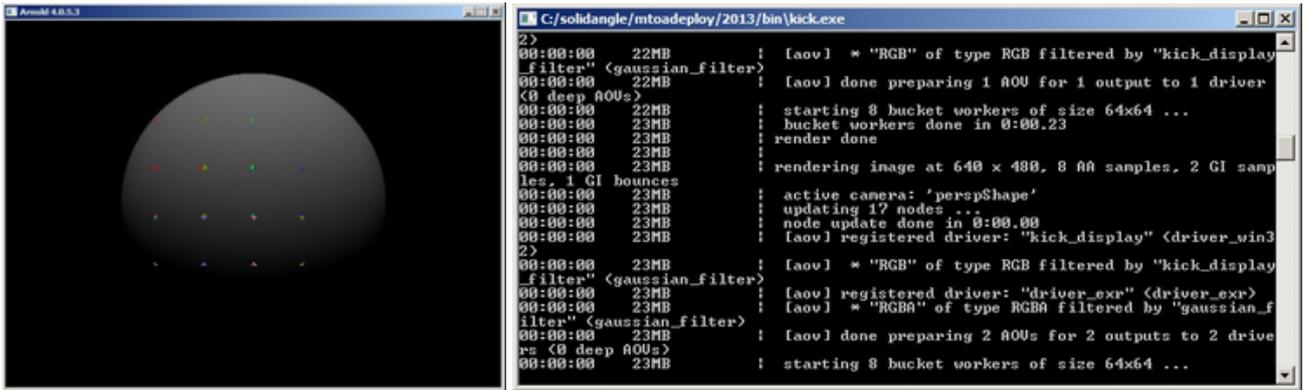
## Render Type

Choose between *Interactive* rendering, *Export Ass*, or *Export Ass and Kick*. These types are described below:

**Interactive:** The default render option for rendering with Arnold which will render using Maya's viewport.

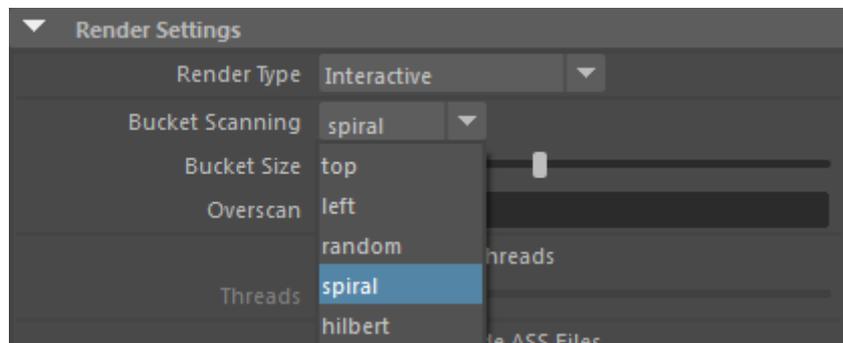
**Export Ass:** This option automatically exports an **.ass** file to the data folder of your current project.

**Export Ass and Kick:** This option exports the scene as an **.ass** file, launches **Kick**, and renders the **.ass** file in a native render view window outside of Maya as shown below:

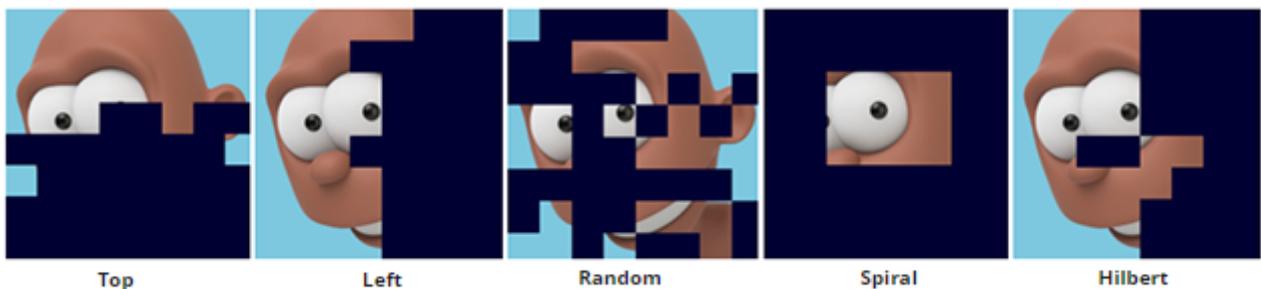


Rendering via Kick

## Bucket Scanning



Specifies the spatial order in which the image buckets (i.e. threads) will be processed. By default, buckets start in the center of the image and proceed outwards in a spiral pattern.



## Bucket Size

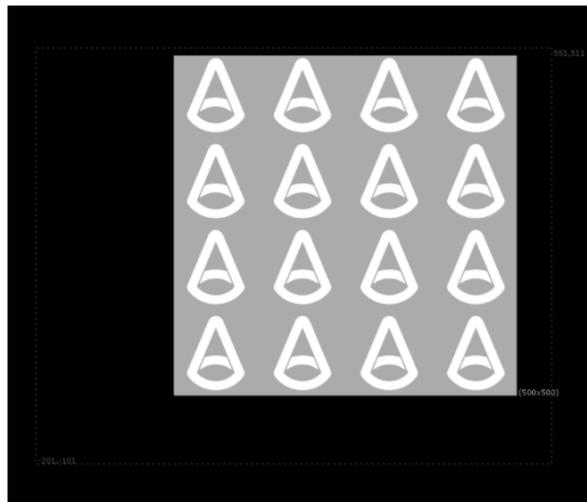
The size of the image buckets. The default size is 64x64 pixels, which is a good compromise; bigger buckets use more memory, while smaller buckets may perform redundant computations and filtering and thus render slower but give initial faster feedback.

## Overscan

Overscan extends the render region beyond the regular image coordinates. It can be either a percentage or a pixel value. You can mix % and pixel values at the same time, however, if you don't specify the unit it will default to pixels. It can have:

- One value such as 10px to specify an equal margin on every side.
- Two values, such as 10px 5px, to specify the top/bottom (first value) and right/left (second value) margin.
- Three values, such as 10px 5px 2px, to specify the top (first value), right/left (second value) and bottom (third value) margin.
- Four values, such as 10px 5px 2px 1px to specify the top, right, bottom, and left margins respectively.

E.g. a ten-pixel overscan for a 640x480 image in all directions can be achieved with options *region\_min\_x*, *region\_min\_y*, *region\_max\_x*, *region\_max\_y* of -10, -10, 649, 489. The example below shows a pixel value of 10, 50, 100, 200 (top, right, bottom, left) in the 'Overscan' field:



Overscan set to 10, 50, 100, 200 and the result viewed in Nuke

**i** There is a way to view the output image of a render in progress. You must do the following:

- Set your output to be tiled EXR, zip is fine.
- Set the bucket scanning method to 'top'.
- Use a viewer like imf\_disp.

**i** Note that overscan will only work with batch rendering and the EXR file format.

### Autodetect Threads

Use an optimal number of render threads based on the number of hardware processing cores.

### **Threads**

If autodetect is disabled, this will let you manually set the number of rendering threads. Negative numbers are also allowed. If specifying 0 threads means to use all cores on a machine, then negative numbers can mean use all but that many cores. For example, threads=-2 means use all but two cores, while threads=2 means only use two cores. This is useful when you want to leave one or two cores for other tasks. One example of this is so that the DCC application can be more responsive while Arnold is rendering in the Render View.

### **Binary-encode ass files**

Specify whether binary encoding is used to compress large float arrays into a more compact ASCII representation, leading to smaller files and faster load times, while still being mostly human-readable. Also, the binary encoding has exact 32-bit precision, whereas without this binary output floating point values are truncated into at most 8 ASCII digits (e.g. 1234.5678). The encoded arrays are indicated by prefixing the array type with "b64".

Note that you can read more about this in the [Exporting to Arnold Scene Source](#) topic.

### **Export Bounding Box (.asstoc)**

Specifies whether the bounding box of the scene is exported in an .asstoc file. This file is used by the stand-in node to draw in the viewport a proper bounding box of the scene inside.

### **Expand Procedurals**

Internally, Arnold creates shape nodes from procedural nodes (usually on demand). 'Expand procedurals' expands the nodes before doing the Ass export. Therefore when saving .ass file, you will get all of the nodes that have been created by the procedural.

### **Expand All Shading Groups**

When enabled, all shading groups are exported (or only the selected ones during export selected), even if they're not assigned to any geometry in the scene. This prevents assignment of shaders to dummy objects.

### **Export Full Paths**

Exports the node names with the full Maya path. For example, pSphere1 | pSphereShape1 will be used instead of pSphereShape1.

## Export Namespace

Determines if the Maya namespaces must be exported or not to the Arnold node names. The mode “root” exports them only once at the root of the full name, as an additional hierarchy. For example, a sphere that was imported in Maya with a namespace “asset” will usually appear in maya as “asset:pSphere1|asset:pSphereShape1”. Exporting in “root” mode will name the Arnold node as “asset/pSphere1/pSphereShape1”

## Export Dag Name

Allows using transform node or shapes (default) names when exporting a scene.

## Export Separator

Determines the separator used for the hierarchy in the Arnold node names. By default, '/' will be used. For example, pSphere1 | pSphereShape1 will be exported by default as /pSphere1/pSphereShape1

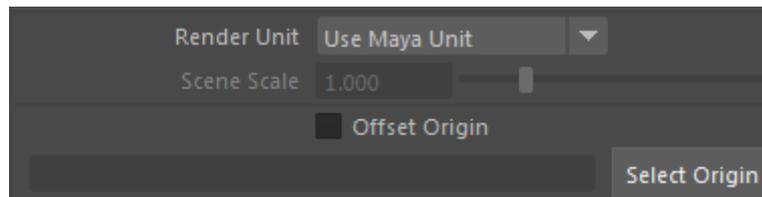
## Export Shading Engine (Legacy)

Exports a Maya-specific MayashadingEngine at the root of all shading trees. This parameter is provided for legacy reasons and will be removed in a future release. Note that previously saved .ass files are still compatible. However, previously exported XGen archives may not render properly.

## Kick Render Flags

This string field can be used to pass parameters to Kick when the render type is set to Export Ass and Kick.

## Render Unit



*Use Maya Unit* - Uses the default Maya unit settings.

*Use Custom Scaling* - Here you can specify a float value that is used to scale the scene.

**Scene Scale** - A global scene scale parameter (set to 1.0 by default).

**Offset Origin** - Select the origin of the coordinate system.



- It is recommended that you model your scene to a real-world scale.
- When exporting scenes to Arnold, the scene units defined in the Render Settings (**System**) are stored in the .ass file header. This allows loading the file in a scene having different scene units, with consistent results. A *stand\_in* loading a .ass file will now appear with a different scale if the current scene units are different from the ones used during export. By default, MtoA uses the scene units defined in the Maya Preferences.

## Callbacks

All of the common callbacks in the Common tab of the Render Settings behave as intended, except Post Render. Post Render callback is executed after translation ends, and MtoA gives back execution to Maya. Since MtoA renders in a different thread from the main thread, the Post Render is always executed before the actual render finishes. If you require a callback that happens exactly after the render, then you must use 'Post Render Frame'.

MtoA also provides an additional set of callbacks for IPR events, which can be found in System / Callbacks. These string fields can be used to add pre/post mel (or python) callbacks specific to IPR steps /stages. To better understand these callbacks, we have to specify Refinement and Step (that are individual renders with increasing AA) in this context. Between scene changes during an IPR session, MtoA goes through a phase called Refinement. Every Refinement contains several render passes, which are called 'Steps' in this context. These callbacks are executed before or after any of these phases. If nothing has changed, MtoA calls Refinement Finished. If something changes in the scene, MtoA stops the Refinement phase (thus calling Refinement Finished) and start a new one.