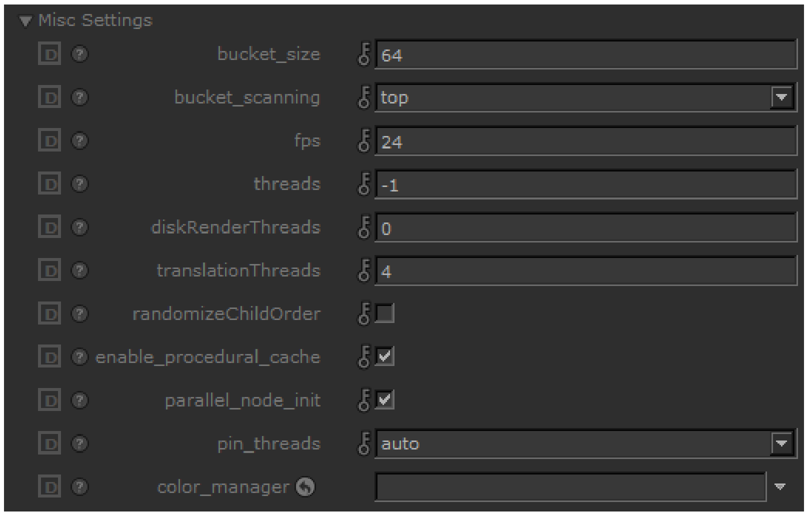
Misc Settings



Misc Settings

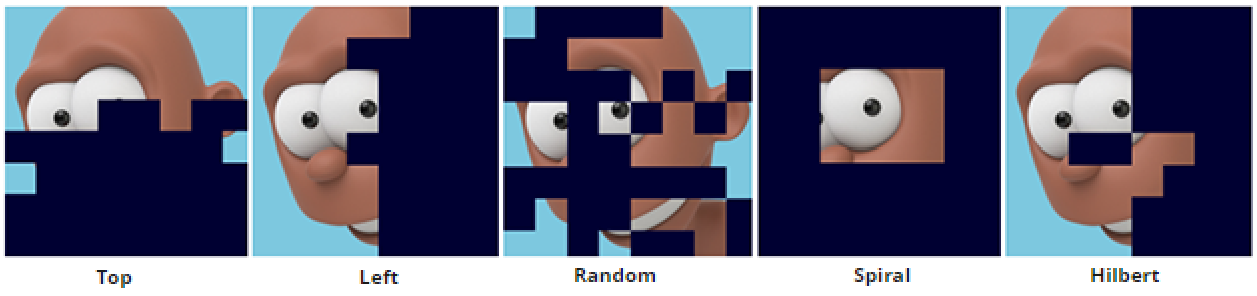
bucket_size	64
bucket_scanning	top
fps	24
threads	-1
diskRenderThreads	0
translationThreads	4
randomizeChildOrder	☐
enable_procedural_cache	✓
parallel_node_init	✓
pin_threads	auto
color_manager	

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.

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.



fps

The frame rate in units of frames per second when this scene was exported (default is 24).

threads

The number of threads used during preview and live rendering.

The number of threads used for rendering. Set it to zero to autodetect and use as many threads as cores in the system. Negative values indicate how many cores not to use, so that -3, for instance, will use 29 threads on a 32 logical core machine. Negative values are useful when you want to reserve some of the CPU for non-Arnold tasks.

Negative numbers are also allowed. If specifying 0 threads means 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 2 cores, while threads=2 means only use 2 cores. This is useful when you want to leave one or two cores for other tasks. One example of this is so that Katana can be more responsive while Arnold is rendering in the Render View.

diskRenderThreads

Similar to "threads" this is the number of threads used for batch/disk rendering. Usually you want to use all available threads. This can be overridden when invoking Katana in batch mode with the "–threads3d=" command line parameter.

translationThreads

This is the number of threads KtoA uses when translating the scene. Usually, a handful of threads suffices to give an advantage and making this number higher may actually slow down scene translation.

randomizeChildOrder

Particularly for batch rendering, when many frames are launched on the farm simultaneously, Katana may request the same resources on the network all at once from many different machines. This can cause bottlenecks on file servers as the same files (such as Alembic assets) are requested at the same time. Randomizing the order of children locations can help alleviate this problem and improve overall render throughput on a render farm. When not in this situation, it doesn't help to activate this, and in fact it carries a small amount of performance overhead at render startup.

enable_procedural_cache

Enables automatic instancing of .ass procedurals.

parallel_node_init

Enables parallel initialization of all scene nodes.

pin_threads

Arnold can pin threads on Linux, so they don't jump between multiple processors. This can improve scalability in modern machines, multiple processors. It can be set to *off, on*, or *auto*. By default is set to *auto*, meaning that if the number of threads is more than half the number of logical cores on the machine, Arnold will pin the threads.

> If client code, for instance, a custom shader, spawn their own threads manually (with pthread_create or similar), these threads will inherit the same thread affinity, which totally breaks the point of spawning threads; in these situations, they can either set options.pin_threads to off, or they can create their threads with the Arnold API AiThreadCreate() which will un-pin the created thread.

color_manager

Connects a color manager node such as OCIO.

arbitrary

This area shows any arbitrary values set on the root location's arnoldGlobalStatements.arbitrary attribute group. Use an AttributeSet node or opscript to add arbitrary attributes, and they will be written out as user data to the Arnold options node.