

# 4.0.10.0

## Milestone 4.0.10

### Enhancements

- **Improved texture blurring for inter-reflections:** We used `sg->Rt`, the ray type, to determine when to blur a texture with `texture_diffuse_blur` and `texture_glossy_blur`. We now use the ray depths instead, in the event the ray type changes after a diffuse or glossy bounce. The new method can reduce the amount of texture I/O in inter-reflections, like a glossy sphere reflecting a mirror sphere. (#3057)
- **Faster ray accel build for icc:** An optimization has been made to the ray acceleration structure code that results in build time speedups of these structures of as much as 15%, although typical speedups might be lower depending on the object being built. Furthermore, this fixes a severe performance regression in the icc compiler versions of Arnold that occurred for many types of CPUs (except Sandy Bridge), and in that case build time is as much as 80% faster. (#3099)
- **ignore\_motion\_blur discards multiple key data:** When the `ignore_motion_blur` option is set to true, or when `-imb` is passed to kick, we now get rid of the extra motion keys which results in memory savings and faster ray tracing time since we can use the simpler static acceleration structure instead of the one for deformation motion blur. This can be useful for debugging and benchmarking the cost of deformation motion blur in a scene. The previous behavior where the other motion keys are kept can be kept by manually specifying the shutter to be open at only a single point in time -- for instance, `-sh 0 0` would render only the first key frame. (#3043)
- **Improved .ass procedural cache:** Cached procedural nodes, i.e. those pointing to the same .ass file on disk, are now directly referenced by one instance node, instead of by a separate instance for each child. This can dramatically reduce the memory footprint in complex scenes, as well as reduce scene loading time. In addition, we have added log stats on how many .ass files have been truly loaded from disk vs found in the cache. Finally, the cache has been extensively and successfully tested in production, so its associated option `enable_procedural_cache` is therefore now enabled by default. (#3054, #3097)
- **Faster resolving of forward references:** In complex scenes made up of many .ass-loading recursive procedurals, resolving forward-referenced symbols like shader names was taking a significant chunk of the initialization time (both pre-, and during rendering, for deferred procedurals). This has been greatly optimized and shouldn't be noticeable any more. (#3051)
- **Faster abort during scene loading:** The renderer will now poll for an abort condition during loading of .ass files and arbitrary procedural nodes, which increases interactivity when working with big procedural/standin networks. (#2568, #3085, #3092)
- **Forced procedural expansion:** A global option `procedural_force_expand` has been added that forces all procedural nodes to be opened before rendering (or before writing to .ass) regardless of whether the user provided bounds for them or not. Note that this effectively makes procedural loading non-deferred, single-threaded and exhaustive; scene loading will generally be slower and use more memory because procedurals will be loaded even when hidden behind other objects. This option is intended mainly as a debugging aid. In addition, a corresponding command-line option for kick has been added: `-forceexpand`. (#3078)
- **Ignore empty/invalid procedurals:** procedural nodes which produce an error before expansion (e.g. missing procedurals, empty dso parameter, errors while loading) or successfully loaded procedurals that generate no geometry (e.g. empty .ass files, procedurals containing only shaders) were being processed as renderable objects with empty or uninitialized bounding boxes, which caused various internal problems and obscure warnings. These procedurals are now more clearly rejected and never make it into the renderable scene, while also resulting in less warnings. (#3053)
- **Improved max warning/error cutoff:** The maximum number of warnings specified (e.g. `kick -nw <n>`) now works per warning type and also now applies to errors. Once a particular type of warning/error has been printed N times (the default has been lowered from 200 to 5 in kick), no more of those messages will be shown but other types of messages will still appear. At the end of the log there will be a list of all printed warnings and errors along with the number of times they occurred. (#3066, #3092)
- **Less performance warnings:** We now issue a performance warning for useless identity matrices only when there are more than 100k of them; less than that is probably not worth reporting. (#3069)
- **Enhanced kick attribute overrides:** Extended the `-set` option for kick to fall back to searching for a node type when a named node is not found. This allows a user to set the same attribute on all nodes of a given type without knowing the node names, which can be very handy, for example `kick -set driver_exr.tiled true`. (#3083, #3093)
- **Deep driver improvements:** Some internal infrastructure work has been done in preparation for future support of deep-image output drivers. Threading efficiency has been increased with the addition of an internal non-locking step that allows output drivers to do CPU-intensive work (such as deep-data compression) in parallel without having to wait on other threads. Also, raw drivers can now correctly access all samples in a deep stack, including the first one which was being overwritten and reused for multiple purposes. (#3056, #3068)
- **AiASSWrite() can write to stdout:** In the same way as `AiASSLoad("-")` reads a scene from stdin, it is now possible to send the scene to stdout with `AiASSWrite("-")`. (#3118)
- **Faster AiASSWrite() for .ass.gz files:** The zlib compression settings are now optimized for speed instead of file size. The new .ass.gz files are slightly bigger (~5%) but are generated ~3x faster. (#3119)

### API additions

- **AI\_MAX\_THREADS:** The maximum number of rendering threads is now exposed via the `AI_MAX_THREADS` macro. (#2038)

### Incompatible changes

- **-resaveop:** Removed obsolete option `-resaveop` from both kick and pykick. The same result can be obtained by preceding the `-resave` option with `-forceexpand`. (#3078)

### Bug fixes

Ticket	Summary	Component	Owner	Priority	Version	Created
#3087	Multi-threading crash due to uninitialized shaders in procedural networks	arnold	angel	blocker	4.0	3 weeks
#3044	light groups override in procedurals not working	arnold	angel	critical	4.0	6 weeks
#3052	Meshes with identical motion keys and unconnected vertices are corrupted	arnold	thiago	critical	4.0	6 weeks
#3080	Crash when loading deferred .ass procedurals with missing node types	arnold	angel	critical	4.0	4 weeks
#3090	AiTraceProbes crashing with procedurals	arnold	thiago	critical	4.0	3 weeks
#3094	Threading deadlock in .ass-based procedurals due to forward reference resolution	arnold	angel	critical	4.0	3 weeks
#3048	Let volume shaders interact with skydome and distant lights	arnold	ramon	major	4.0	6 weeks
#3056	First deep AOV sample is being overwritten	arnold	ramon	major	4.0	6 weeks
#3065	artifacts in volume scattering with textured spotlight gobos in front of geometry	arnold	oscar	major	4.0	5 weeks
#3072	Fix support for linear basis in curves	arnold	ramon	major	4.0	4 weeks
#3073	Displaced bounding box computation not thread safe	arnold	ramon	major	4.0	4 weeks
#3077	ambient occlusion should be visible to indirect diffuse rays	arnold	marcos	major	4.0	4 weeks
#3082	kick returns error code "1" after rendering if window is open	kick	thiago	major	4.0	4 weeks
#3084	Warning about unsuccessful DLL loading does not give DLL name	arnold	thiago	major	4.0	4 weeks
#3086	visible/unique triangle stats are not correct for nested procedurals	arnold	thiago	major	4.0	3 weeks
#3097	remove excessive "added/round" messages in procedural cache	arnold	oscar	major	4.0	3 weeks
#3104	deferred .ass loading time stats are not thread-safe	arnold	marcos	major	4.0	2 weeks
#3106	.ass-based procedurals should filter out more node types	arnold	marcos	major	4.0	2 weeks
#3110	EXR append does not work when using additional channels besides RGBA	arnold	ramon	major	4.0	10 days
#3119	slow writing of compressed .ass.gz files	arnold	angel	major	4.0	36 hours
#3061	Minor inconsistencies in node memory stats	arnold	angel	minor	4.0	5 weeks
#3089	rephrase warning for bad assignment of atmosphere shader	arnold	marcos	minor	4.0	3 weeks
#3093	kick crashes on windows when using an invalid -set option	kick	oscar	minor	4.0	3 weeks
#3115	remove procedural indentation in log messages	arnold	marcos	minor	4.0	4 days
#3049	Warn when loading procedurals with empty "dso" parameter	arnold	oscar	trivial	4.0	6 weeks
#3081	kick doesn't report the right arnold version	kick	oscar	trivial	4.0	4 weeks