

4.0.8.0

Milestone 4.0.8

Enhancements

- **Reduced texture I/O:** Ray differentials, used to calculate texture sampling footprints, have been improved resulting in generally larger filter sizes matching the requested visible detail. Texture sampling will now choose less detailed MIP levels when possible and will generally load less data from disk and do less work. In particular, ray differentials have been improved for mirror reflection rays (of type `AI_RAY_REFLECTED`), refraction rays (in the standard shader only), ray-traced SSS rays, and finally for camera rays during the first rough passes of progressive rendering at negative AA settings, where textures will be blurred to match the size of the blocks of pixels and help decrease startup time. We have seen dramatic speedups in scenes with large numbers of high-resolution texture maps because of the significantly reduced disk I/O. (#2979, #3017, #3022, #3028)
- **Faster bump mapping:** Autobump and bump mapping are now cheaper to compute than before due to there being 25% less shader evaluations per call. (#2909)
- **Faster ray accel build:** It now takes about 25% less time to build the ray acceleration structures. Memory usage and ray tracing speed are unaffected. (#3008)
- **Thread affinity:** Threads can now be pinned to cores on Linux. This can improve scalability in modern machines with 16 or 24+ threads and, in the case of ray acceleration structure builds, it can occasionally result in as much as a 3x speedup when all the cores are being used (which is the default in Arnold, `options.threads = 0` or `-t 0` in kick). The default setting is auto where thread pinning is only enabled if more than half the logical cores are being used. This can also be manually set to always on or off by setting the new global option `options.pin_threads` to on, off, or auto. Note that, if client code, for instance a custom shader, spawns 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. (#2733)
- **Auto-instancing of .ass procedurals:** Thanks to a new built-in cache, procedural nodes whose `dso` parameter points to an `.ass` file can be automatically cached for future reuse; subsequent procedurals trying to load the same `.ass` file will instance the previously loaded geometry, potentially saving tons of both memory and disk I/O. Because of the experimental nature of this feature, we have opted for disabling it by default and let those users that need it manually enable it with the global option `enable_procedural_cache`. (#2937)
- **AiOrenNayarIntegrate() caching for shader networks:** Just like we already do with `AiIndirectDiffuse()`, multiple calls to `AiOrenNayarIntegrate()` with the same arguments within a shader network will now be optimized so that the result is only calculated once and then reused. (#3004)
- **bumpdiff mode in the utility shader:** The utility shader has a new value available for the `color_mode` parameter: `bumpdiff`. This will show how far the bump and autobump normals vary from the base smooth-shaded normals as a heatmap (blue is the same, going through green to red as varying up to 90 degrees away). This is useful for debugging the balance between subdivision iterations with displacement vs autobump making up for the rest. (#3002)
- **Log file improvements and cleanups:** A number of minor annoyances in `.log` files have been fixed, and a number of messages have been made more helpful. For example: the "ray counts" section now mention the number of rays fired from SSS pointclouds which were previously not reported; the "render time" section now reports the time it takes to build the importance sampling tables for the `skydome_light` and `quad_light` nodes; the "subdivision done" message now more clearly reports the initial number of polygons and the final number of subdivided quads; the "displacement done" message now reports the number of vertices that were displaced; and the "loading metadata file" message now only appears with debug verbosity. (#3029)
- **Updated pykick:** The Python version of kick has been updated so that it now has all the recent command-line options that were added in the C version of kick. (#2984)
- **Static RLM linking in Windows:** To make things more consistent with Linux and OSX, we now statically link the RLM library in Windows too; we don't distribute the files `rlm*.dll` anymore. (#2982)

API additions

- **AiUserDataGetNode():** User-data queries of type `NODE` are now supported. Recall that the `NODE` type is very similar to the `POINTER` type, with the only difference that an element of `NODE` type is guaranteed to point to an Arnold `AtNode` object rather than being a pointer to an arbitrary memory location. (#2771)

Incompatible changes

- **Changed AiLicenseGetInfo():** The `AiLicenseGetInfo` API has been changed so that it doesn't use `std::vector`, which is known to cause problems across compilers and compiler settings. The new safer API is: (#2826)

```
AI_API int AiLicenseGetInfo(AtLicenseInfo*& licenses, unsigned int& n)
```

- **Removed ARNOLD_DISPLACEMENT_DERIVS_HACK:** This undocumented legacy environment variable is no longer necessary. (#2586)

Bug fixes

Ticket	Summary	Component	Owner	Priority	Version	Created
#2953	shading artifacts in procedural networks with instances of procedurals	arnold	oscar	critical	4.0	6 weeks
#2842	Arnold crashing on unexpected camera shader	arnold	oscar	major	4.0	4 months
#2979	Compute ray direction differentials for reflection	arnold	mike	major	4.0	4 weeks
#2983	faceted artifacts in raytraced SSS	arnold	alan	major	4.0	4 weeks
#2985	Crash in AiEnd when using AiRender after an AiASSWrite	arnold	oscar	major	4.0	4 weeks
#2987	Crash when aborting on license fail	arnold	oscar	major	4.0	4 weeks
#2989	AiSampler() crashes when sample count is 0	arnold	alan	major	4.0	4 weeks
#2993	Tile boundaries incorrect for udim tag	arnold	ramon	major	4.0	3 weeks
#2994	OS X compiled shaders do not load	arnold	ramon	major	4.0	3 weeks
#2997	Distant lights do not cast shadow rays in volumes	arnold	ramon	major	4.0	3 weeks
#2998	random corruption of rays	arnold	alan	major	4.0	3 weeks
#3000	Improve numerical robustness of texture derivatives	arnold	thiago	major	4.0	3 weeks
#3014	missing Doxygen docs for AiASSWrite/AiASSLoad	arnold	marcos	major	4.0	2 weeks
#3016	broken normal smoothing for NURBS primitives	arnold	oscar	major	4.0	2 weeks
#3017	Compute ray direction differentials for refraction in standard shader	arnold	mike	major	4.0	13 days
#3021	correlation artifact in indirectly sampled area lights	arnold	alan	major	4.0	9 days
#3027	"ray counts" stats are missing SSS rays	arnold	mike	major	4.0	5 days
#2977	Memory leak in kick when setting the outputs	kick	oscar	minor	4.0	4 weeks
#2996	Texture Errors Should Report Filename	arnold	ramon	minor	4.0	3 weeks