

4.2.3.0

Milestone 4.2.3.0

Enhancements

- **Faster curves:** Several optimizations have been done on the ray/curve primitive test giving up to 40% faster hair performance. In the process, the `max_subdivs` attribute of the curves node has been removed. In addition, the internals of the curves primitive acceleration structure have been tweaked, giving up to 20% memory savings with 10% faster preprocessing. (#4386, #4411, #4422, #4433, #4437, #4441, #4454)
- **Faster volumes:** Shadow rays through both bounded and atmospheric volumes are now faster. If image artifacts are noticed, decrease the volume's `step_size`. Speedups are scene-dependent, but we've seen more than a 2x speedup in many of our tests, with little to no quality degradation. (#4377, #4383, #4429)
- **Up to 128 threads:** Arnold now allows for rendering with up to 128 threads (increased from 64). Shaders that make use of `AI_MAX_THREADS` will need to be recompiled otherwise crashes may occur in upcoming machines with more than 64 threads. If you do not have machines with more than 64 threads, there is no need to recompile any shaders. In addition, like in Linux, Windows can now make use of the existing `options.pin_threads`. (#4337, #4330)
- **Parallel node initialization/update:** Initialization and update of scene nodes can now be multi-threaded. This can significantly lower the scene preparation time in complex scenes with many objects, shaders or lights. This extends the parallel procedural initialization implemented in the previous 4.2.2 release. The previous `parallel_procedural_init` option has been renamed to `parallel_node_init` because it now applies to all nodes, not just procedural nodes. For now, this option is disabled by default, to avoid issues with existing custom shaders or procedurals that may have hidden implicit dependencies between nodes, something which is not currently compatible with multi-threading. Please try this out and report any problems. (#4329, #4358)
- **Upgrade to latest compiler:** OS X and Linux use a newer version of clang. This results in slightly faster code. Shaders and plugins can safely continue to be built with older compilers. (#4354, #4356)
- **User normals in linear subdiv and displacement:** User-specified polymesh vertex normals are now preserved and respected during displacement, and also when using linear subdivision. For cases where Catmull-Clark subdivision is not needed, this allows sharp edges to remain sharp when increasing mesh density for detailed displacement. Catmull-Clark subdivision still warns about the presence of user normals and throws them away in favor of the smoothed limit normals. It is still usually preferable to use Catmull-Clark subdivision with the new fractional creasing instead of linear subdiv and user normals, however. (#4306)
- **Per-node 'active' flag:** Scene nodes can now be enabled or disabled for rendering (this is a debug feature that is not preserved in `.ass` files). For now only lights, shaders and objects can be enabled or disabled. (#4150)
- **Kick display of individual color channels:** Pressing `shift + r, g, b, a` in interactive kick now displays the corresponding color channels. `shift + c` resets to RGBA display. (#4366)
- **Autobump support for meshes with no UVs:** The autobump detail enhancement now works for procedurally displaced polymeshes even when there are no UV texture coordinates. (#4453)

API additions

- **Per-node 'active' flag:** New API functions are provided to set and query the new flag: (#4150)

```
void AiNodeSetDisabled(AtNode* node, bool disabled);
bool AiNodeIsDisabled(AtNode* node);
```

Incompatible changes

- **Raised minimum Windows version to Windows 7:** We no longer support Windows versions before Windows 7 and Windows Server 2008 R2. (#2871, #4330)
- **Atmosphere shaders and `sg->Vo`:** Custom atmospheric shaders (those attached to `options.atmosphere`) are now required to fill out `sg->Vo` with the volume's additive color in the same way that volume shaders already do. For instance, for a fog shader whose atmosphere contribution scatters `fog_radiance` towards the ray origin and attenuates the background according to `fog_opacity`, things should be set in a similar way to this: (#4389)

```
sg->Vo = fog_radiance;
sg->out.RGB = sg->Ci * (AI_RGB_WHITE - fog_opacity) + fog_radiance;
```

- **AiTrace() for shadow rays:** If `AiTrace()` is passed a ray of type `AI_RAY_SHADOW`, it will now more quickly compute the sample opacity, but it will no longer return any other sample information. If that is required, `AiTraceProbe()` can be used. Note that, previously, the sample information it provided was not for the first hit, but rather a random hit. So aside from the opacity, this never provided reliable sample information. (#4377)
- **Full-frame spherical and cylindrical cameras:** Spherical and cylindrical cameras now maintain their FOV vertically regardless of frame and pixel aspect ratio. For spherical cameras this means that the render will always reach all the way to the poles vertically. For cylindrical cameras `vertical_fov` will now be correctly respected. Tweaking `options.aspect_ratio` is not needed anymore to get the correct FOV. This change does not affect custom cameras. (#4398)
- **Removed `enable_aov_composition`:** The global option `enable_aov_composition` no longer had any effect on performance so it has been removed and now compositing of semi-transparent AOVs is always performed correctly. (#4404)
- **RGBA AOV composition:** The A channel of RGBA AOVs is now fully composed as an additional alpha channel per AOV. Previously no compositing would happen for this channel, and just the value from the first surface hit would be reported (irrespective of the particular AOV being set or not) (#4408)

- **8-bit output when alpha is 0:** A bug has been fixed that was preventing correct file output if A was set to 0 and gamma correction was used. Before this bug was fixed, the output would have been (0,0,0,0) instead of the correct (R, G, B, 0). (#4420)
- **AI_MAX_THREADS increased to 128:** AI_MAX_THREADS has been increased to 128, so any code that relies on it and needs to support more than 64 threads must be recompiled. (#4337)
- **Removed legacy cubic curve support:** The unused cubic bases hermite and power in the curves node have been removed. (#4441)
- **Removed curves.max_subdivs:** The max_subdivs attribute of the curves node has been hardcoded to 6 (the previous default) and removed. The only reason to tweak this parameter was to get a tiny speedup at the cost of lower quality, faceted curves; this tiny speedup is now dwarfed by the more significant ray/curves speed gains found in this release.
- **Autobump enabled for meshes with no UVs:** Previously polymesh.disp_autobump true was ignored on polymeshes without UV texture coordinates. Now, autobump will be correctly applied which might cause a change in look (improved displacement detail) for that specific case. (#4453)

Bug fixes

Ticket	Summary
#4393	main thread was pinned to a single core
#2871	Windows CPU detection code is wrong for high-end machines
#4330	Windows is unable to use more than 64 logical cores
#4355	Compute correct differentials when persp_camera.uv_remap is used
#4359	memory leak in AiTraceProbe()
#4367	Matte override on ginstance is ignored
#4378	streak artifact in quad lights
#4385	Kick display issue with fast cropped renders
#4387	excessive volumetric noise in fast motion-blurred quad lights
#4390	Skydome light crash and IPR issues with null energy in linked color
#4394	Kick display window issues with old or incomplete OpenGL drivers
#4398	Spherical and cylindrical cameras incorrect cropping for non square frames
#4402	User data of type array crashes when accessed for displacement
#4403	Kick displays a black window when returning from the lock screen
#4408	RGBA AOV alpha component should composite using the global alpha
#4409	Bright dots in fog at the horizon line
#4410	random flat areas in bump mapping
#4420	Unable to output purely additive pixels with gamma
#4428	Polymeshes with zero area faces can have wrong user data for some faces
#4430	Curves incorrectly clipped in render in some cases
#4442	Allow for motion blurred FOV on custom cameras tagged with is_perspective
#4448	Compositing issues with fully transparent samples
#4452	Bad normals with autobump and vector displacement
#4457	Crash with invalid time_samples parameter
#4439	pykick doesn't understand negative AA samples on command line
#4462	Metadata files with Windows EOL are not correctly parsed in OSX