

5.4.0.0

31 Jul 2019

Enhancements

- **Improved progressive and adaptive sampling:** Better progressive sampling patterns lead to increased noise convergence rate and faster adaptive sampling termination. (#8447)
- **clip_geo shader:** The new `clip_geo` shader will clip out all geometry against the shapes it has been assigned to. You can limit the objects that are affected with trace sets, and optionally choose a specific shader for the intersection surfaces. Note that self-intersecting geometry or intersecting clipping objects can cause artifacts. (#7274)
- **AOV read shaders:** The new `aov_read_*` shaders let you access built-in AOVs in the AOV shader context or custom AOVs in the regular shading context. They can be useful to reformat built-in AOVs, for example to output normal maps derived from the built-in `N` AOVs when baking to texture. (#7787)
- **Alembic Layering:** The `alembic` procedural now supports Alembic layering, where extra Alembic files specified via the new `layers` parameter will override properties on top of the Alembic file. (#8467)
- **Alembic Arnold Properties:** The `alembic` procedural will now set Arnold parameters from `arbGeomParams` if they match any Arnold parameter name and type. (#7577)
- **Alembic Material Attribute:** The `alembic` procedural has a new `material_attribute` optional parameter that defines the Alembic property that contains the material names, which will be used to create the per polygon material assignments. (#8535)
- **Support for the MaterialX standard library:** The `materialx` operator now supports look development using nodes and node graphs in the MaterialX standard library. Shading models defined using the PBR library are generated using the MaterialX Arnold OSL shader generator. (#8453)
- **Disable assignment expressions:** Assignments in the `set_parameter` operator can be now be individually disabled. (#7182)
- **Individual parallel init selection for procedurals:** You can now tag individual procedural types for serial or parallel initialization. All built-in procedurals are tagged for parallel initialization, but custom procedurals are serial by default, unless their `parallel_init` boolean metadata has been set. This is useful to tag non-thread safe procedural without disabling parallel initialization altogether. (#7271)
- **Autodesk analytics program:** Arnold can now optionally send anonymous statistics data back to Autodesk for analysis. This will help us optimize Arnold for the workflows and systems that are commonly used. This service can be opted-in either by calling the `AiPromptADPDialog()` API, through the kick `-ADP` flag, or through other Autodesk applications that support this, such as Maya. For the moment this is only available for OSX. (#8433)
- **oiitool:** `oiitool` is now distributed with Arnold. (#8501)
- **Improved round corners:** `round_corners` no longer generates non physical normals, this prevents excessive darkening near objects' silhouettes. (#8365)
- **Faster individual node destruction:** Destroying individual nodes with `AiNodeDestroy()` is now substantially faster in scenes with many global-scope nodes. Individually destroying 500k ginstance nodes went from 50s to about 2s. (#8343)
- **Faster scene creation and destruction on Windows:** Poor Windows performance when creating and destroying scenes has been greatly improved so that performance is almost as fast as Linux and OSX. For instance, the "releasing resources" step at the end of a render with 29k curves objects went from 45s to 4s. (#4399)

GPU enhancements

- **Open Shading Language:** Initial support for OSL has been added to the GPU. Like on the CPU you can mix and match OSL and compiled shaders. Note that not all closure and shading ops are supported yet, refer to the [supported features and known limitations docs](#) for details. (#5975)
- **OpenVDB Volumes:** Initial OpenVDB support, note that this initial implementation is likely to use more memory on the GPU than on the CPU. (#7051)
- **Volume displacement:** The volume displacement and padding logic has been implemented on GPU. (#7606)
- **Lights compatibility :** Added support for light portals, cylinder lights, disk lights, and low light threshold. (#8531, #8540, #8557, #8551)
- **Standard Surface compatibility:** Added support for Randomwalk v2 SSS and `transmission_dispersion`. (#8253, #8253)
- **Less noise in indirect samples:** Better stratification now reduces the noise in indirect samples, for example with refractive surfaces. (#8615)
- **Better GPU cache pre-population ETA:** More accurate GPU cache prepopulation time remaining estimate. (#8361)

API additions

- **Earlier listing of built-in AOVs:** Built-in AOVs and LPEs are now also listed by the `AtAOVIterator`, no longer requiring an `AiRender()` call. (#8431)
- **New functions for custom procedurals:** Custom procedurals can now define new functions `procedural_update` and `procedural_finish` (equivalent to `node_update` and `node_finish`). Those custom functions (if present) will be executed during scene update and during procedural destruction, respectively (#7961).
- **AiPromptADPDialog():** Call `AiPromptADPDialog()` in order to opt-in or out of the Autodesk analytics program (ADP). Currently ADP is only implemented for OSX, so for other platforms this is a no-op. There is a known limitation that only allows this function to work if the Arnold libraries are in the same path as the host process. (#8433)

Incompatible changes

- **Trace sets:** a trace set is now an optional named set of objects to ray trace against or to exclude. Geometry with no trace set is no longer hit by all inclusive trace sets. The one exception are procedural, which are always traced against when they have no trace set, so that their contents can be considered. (#8358)
- **complex_ior shader deprecated:** `complex_ior` is only usable with the deprecated `standard` shader and has been deprecated too. (#8457)
- **Object transform removed from Alembic procedural:** The `alembic` procedural node's `object_transform` matrix parameter is no longer used and has been removed, and should instead be baked into the procedural's transform if a matrix offset is required. (#8148)
- **Crashes with mismatched AiMalloc/AiFree:** A gentle reminder that `AiMalloc` needs to be paired with `AiFree` or crashes can occur. This was already the case on Linux and now this is also the case on Windows. (#4399)
- **Adaptive sampling criterion removed:** Removed the `AA_adaptive_criterion` option, we now use the best criterion. (#8556)
- **Random token is now a function:** The `random` token in operator expression assignments is now a function and needs trailing brackets: `random()`. (#7761)
- **JSON log file periods:** The JSON log file was using periods for ".ass parsing" and "accel. building" which can cause issues with some JSON readers. We've removed the periods and incremented the JSON log version to 3. (#8357)

Bug fixes

#8177 [Alembic] `alembic_proc` is leaking public symbols

#8591 [Alembic] Allow environment variables in filename and layers

#8057 [Alembic] Allow semi-colon separator in `procedural_searchpath`

#8494 [Alembic] Crash with subdivided geometry that contains only edge creases

#8487 [Alembic] "curves:mode" and "curves:basis" user parameter overrides not enforced

#8529 [Alembic] Disambiguate identical meshes with different UVs for auto instancing

#8548 [Alembic] Error apply Alembic properties to Arnold array parameters

#8500 [Alembic] Incorrect non-indexed UVs

#8567 [Alembic] Position arrays expanded as float arrays

#8505 [Alembic] Shader override fails with `make_instance` on macOS

#7367 [Alembic] Should warn against converting curve basis to bezier

#8434 Closures are incorrectly merged when tracesets, indirect/direct weights, labels, extra depth/samples are different

#8490 Crash while subdividing mesh with large custom indexed attribute

#8358 Geometry with no trace set should not always be included

#7899 [GPU] Arnold GPU crashes Maya with ARV active and turning off motion blur

#8508 [GPU] Clarify the error message when the OptiX wrapper fails to find the runtime

#8192 [GPU] Crashes and update issues with lights and cameras in Gaffer

#8594 [GPU] Crash when destroying nodes in parallel

#8658 [GPU] Crash with ignore shader and volume displacement

#8587 [GPU] Memory leak and OptiX assert when interrupting the GPU prepare stage

#5975 [GPU] OSL on GPU

#8615 [GPU] Poor stratification of indirect light samples

#8253 [GPU] Port `randomwalk_v2`

#8536 [GPU] Port `volume_sample_*` shaders for SDF rendering

#8540 [GPU] Support cylinder lights

#8557 [GPU] Support disk lights

#8531 [GPU] Support light portals

#8551 [GPU] Support `low_light_threshold`

#7373 [GPU] Support render regions beyond image bounds on the GPU

#7606 [GPU] Volume padding and displacement logic

#8484 [GPU] Wavelength sampling in integrator

#8357 JSON log format cleanups

#8664 MaterialX: Erroneous value processing for existing generated shading models

#7670 [Operators] Assignment expressions should automatically resize array parameters

#8554 [Operators] Crash during reset when deleting an operator disconnected from a procedural

#8507 [Operators] Crash when changing target of include_graph operator

#7761 [Operators] Ensure deterministic results when using then random function in expressions

#8509 [Operators] Random crash/hang when creating a secondary universe from an operator

#8489 [Operators] Slow cook of operator network attached to multiple Alembic nodes

#8456 Precision issue in quad light volume bounds computation

#8546 Race condition when creating secondary universes from multiple threads

#8460 RLM accumulates "All licenses in use (-22)" messages