

# 6.0.1.0

10 Dec 2019

These release notes combine Arnold 6.0.0.0 and 6.0.10 since 6.0.0.0 was an internal release for the Maya 2020 installer.

## Enhancements

- **Dielectric microfacet multiple scattering:** Rough dielectrics are now energy preserving by accounting for multiple scattering between microfacets for both reflection and refraction, avoiding the energy loss of the previous implementation. Disabling the global option `enable_microfacet_multiscatter` will restore the previous look. (#8208)
- **Improved rough thin-wall transmission in standard\_surface:** Refractions in thin-walled mode now appear blurry with non-zero roughness. We now consider all orders of interreflection between two interfaces of the thin pane, which leads to more reflection and less transmission than the prior implementation. Rough thin-wall dielectrics assume the same roughness on both sides and the resulting BSDF is energy preserving. Note that the noise might be increased in some scenes at lower AA and roughness, that can be mitigated by increasing `GI_transmission_samples`. We are looking in to addressing this caveat. (#8947, #8180, #8947)
- **Improved roughness mapping of the Oren-Nayar diffuse BRDF:** The Oren-Nayar roughness parameter has been remapped internally so that values close to 1 no longer result in excessive darkening. This change also affects the `standard_surface` and `car_paint` shaders. Scaling the diffuse/base roughness by a factor of 4.44 will restore the previous look. (#8839)
- **physical\_sky now goes below horizon:** The physical sky shader will now extend the sky color at the horizon all the way down to the bottom pole. This should remove the black band under the horizon when not using an infinite ground plane. (#3612)
- **Progressive rendering improvements:** The memory consumption in progressive renders has been greatly reduced, and is now up to 10x smaller. The filtering performance has also been improved. Note that these improvements currently require all outputs to either use box or closest filtering and are not compatible with deep drivers. (#9067, #9094, #9091)
- **Reduced adaptive sampling overhead:** The performance overhead from adaptive sampling with large max AA has been substantially reduced. (#8347)
- **Faster creased subdivision:** The subdivision of surfaces with hard creases is now multithreaded, including in adaptive mode. (#4336)
- **Improved numerical precision when rendering faraway objects:** Numerical precision issues causing artifact in shadows have been greatly reduced, with objects close to the origin now looking similar to objects placed 200k units away. (#8802)
- **New aov\_write\_vector shader:** This enables the writing of vector values into a `vector` typed AOV, for example for recording positional values. These would previously have been clamped when using `rgb` typed AOVs. (#8910)
- **More accurate albedo AOVs:** Albedo AOVs now correspond more closely to the true albedos of the material's BSDFs. (#7479)
- **Alembic: Increased max edges for polymeshes:** Users can now render polymeshes with more than 255 edges per face. (#9106)
- **Alembic: Faster curve generation:** Generating curves in the alembic procedural is now up to 3x faster. (#9044)
- **Skip RGBA denoising:** `noice` now accepts `-ignore_rgba` or `-irgba` to skip denoising of RGBA even if it's present. (#8732)
- **New single-user licensing mode:** Now the user can sign-in with the Autodesk ID for being authorized, buying subscriptions, etc. Single-user licensing mode requires installation of the *Autodesk Licensing Service* and *Single Sign On* components that can be installed from the **Arnold License Manager** tool if necessary. (#7455)
- **Arnold License Manager:** The new Arnold License Manager tool, located in the bin folder, allows to easily configure Arnold licensing through a graphical user interface. Users can select and change license type (e.g. Single-user, Network or RLM), sign in to their Autodesk Account, specify license server addresses for network licensing, etc. You can still configure licensing through environment variables, such as `ARNOLD_LICENSE_ORDER`, `solidangle_LICENSE`, `ADSKFLEX_LICENSE_FILE`, in which case they override the settings in the Arnold License Manager configuration file. (#8805)
- **New ARNOLD\_LICENSE\_ORDER environment variable:** The `ARNOLD_LICENSE_ORDER` environment variable can be set to contain a comma separated string made of the tokens `rlm`, `network`, `user` or `none`. This specifies the licensing systems to use when checking out a license, and in which order. The deprecated `ARNOLD_LICENSE_MANAGER` environment variable will be used if `ARNOLD_LICENSE_ORDER` is not set. The default order is `rlm, network, user`.  
For example, to alter the default order, and use first Autodesk Network Licensing (`network`), and if it fails use RLM (`rlm`):

```
$ export ARNOLD_LICENSE_ORDER=network,rlm
```

To use the new *single-user* model only:

```
$ export ARNOLD_LICENSE_ORDER=user
```

To disable all license managers (you'll always get watermarks!):

```
$ export ARNOLD_LICENSE_ORDER=none
```

- **OCIO roles:** Roles can now be listed with the color manager API by querying color spaces available in the `Role` (OCIO) family. This makes it possible to build UIs that list all known roles. (#6347)
- **Autodesk analytics program available for Linux:** We have extended support for the Autodesk analytics program to Linux. It is disabled by default unless another Autodesk product has opted in. (#8783)
- **OpenImageIO 2.1.4:** OIIO is upgraded to 2.1.4. (#8297)

## GPU enhancements

- Please visit [Getting Started With Arnold GPU](#) and [Arnold GPU FAQ](#)
- Check the compatibility of your system on [Supported GPUs](#)
- Review the [Supported Features and Known Limitations](#)

- **GPU no longer in beta:** Since the majority of the CPU features are now supported on the GPU, with lots of stability fixes and a good match between the CPU and GPU results, we're boldly removing the beta tag!
- **On-demand texture loading:** Textures are now loaded on-demand instead of at the start of the render. This can improve memory usage and time to first pixel if this allows us to read fewer textures. At the moment when we read a texture, we read all data of the required mip-level of that texture, so memory usage can still be higher than on CPU. (#8298, #8674)
- **Texture attribute tag support:** As part of the new on-demand texture loading support, the attribute tagging syntax in the image node is now supported, which allows texture filenames to be constructed at render-time from the user data on the intersected shape. (#8298)
- **UV Camera support:** The `uv_camera` used for baking textures is now supported on the GPU. (#8896)
- **Light filters support:** All light filters are now supported on the GPU, i.e. `light_decay`, `gobo`, `light_blocker` and `barndoor`. (#8588)
- **uv\_remap on cameras:** The `uv_remap` parameter on `persp_camera` is now supported on the GPU. (#9107)
- **filtermap on cameras:** The `filtermap` parameter on `persp_camera` is now supported on the GPU. (#9083)
- **merge\_shader on VR cameras:** The `merge_shader` parameter on the `vr_camera` is now supported on the GPU. (#9108)
- **OSL transforms:** OSL transforms are now supported on the GPU. (#8766)
- **Shadow matte shader:** An implementation of `shadow_matte` has been added to the GPU renderer. This initial version has a number of limitations currently, including: no support for indirect lighting, no AOV generation, and possibly incorrect self-reflections in shadow-matte objects. (#8840)
- **Faster procedural updates:** Procedural children are now updated in parallel on the GPU, bringing a 2-3x speedup on scenes with procedurals generating a large number of nodes. (#8681)
- **Reduced BVH memory used by geometry:** BVH memory used by geometry is reduced. For large meshes we've seen it roughly cut in half. (#7917)
- **Improved GPU Optix cache:** The GPU Optix cache can now be up to 10GB in size and is shared across all GPU configurations. (#8242)
- **Improved interactivity in IPR:** Updating the scene should be more fluid even when rendering large scenes with on-demand texture loading. (#7296, #9014)
- **Improved startup times:** Time to first pixel has been reduced by 20% in some scenes. (#9175)
- **Faster cache prepopulation:** The GPU shader cache population should be approximately twice faster, and shader recompilations triggered less often. (#8829)
- **Multi-GPU rendering without NVLink:** Rendering on multiple GPUs without NVLink should now scale much better. (#9109)

## API additions

- **AiDeviceTypeIsSupported():** The `AiDeviceTypeIsSupported()` API has been introduced to expose an interface to query if a device type is supported on a given system independently from calling `AiRender()`. This API will check your devices and perform any driver or system checks to return if the device is supported and reason for the given outcome. This API can be useful to call when tailoring a UI depending on if the system can support certain devices. (#8736)
- **AI\_RENDER\_UPDATE\_FINISHED:** When using a render callback, when the final pass is done it will send an update of type `AI_RENDER_UPDATE_FINISHED` instead of `AI_RENDER_UPDATE_AFTER_PASS`. (#8818)
- **AtRenderUpdateInfo:** Render callbacks now receive a pointer to `AtRenderUpdateInfo` which has extra pass and sample information. These values are valid for all updates except for interrupts and errors. (#8818)
- **AiRenderAddInteractiveOutput(uint32\_t):** During IPR rendering there can now be more than one interactive output set, meaning for fast and early IPR passes all of those outputs will still be updated and have pixels sent to their respective drivers. Companion API functions `AiRenderRemoveInteractiveOutput(uint32_t)`, `AiRenderIsInteractiveOutput(uint32_t)`, and `AiRende`

`rRemoveAllInteractiveOutputs()` are also available to aid in management of the list of interactive outputs. (#8818)

- **AiLightIESLoad()** will load and parse an IES file and return its maximum intensity and an array of normalized intensities in lat-long format. This is intended for UI manipulators or thumbnails. (#8875)
- **AiBSDFAlbedo()** returns the (directional) albedo of a given BSDF. Can be used in shaders to e.g. layer a specular BRDF over a diffuse one in an energy-preserving way. (#7479)
- **progressive\_max\_pass\_seconds render hint:** In the rendering API, the integer render hint `progressive_max_pass_seconds` is available, which when blocky progressive render passes are slower than this number of seconds it will skip ahead to the pass with AA samples of 1, or to the final AA pass for GPU or regular progressive renders. A value of zero disables the time limit, and that is the default. (#8738)
- **AiProceduralViewport():** Utility API for extracting a proxy representation of procedural contents to a secondary universe, for viewport display purposes. It calls a new (optional) procedural function `procedural_viewport` which can be made to return proxy/basic geometry based on the arguments given. `AiProceduralViewport()` can make use of the new data type `AtParamValueMap` to feed arbitrary parameters to procedurals. The native Arnold procedurals for `.ass`, `.obj`, `.ply` and Alembic now implement this API. (#9110, #9114)
- **AiMicrofacetThinWallRefractionBSDF():** A BSDF that models the transmission of a thin slab of glass. (#8180)
- **AiParamValueMap():** An associative container for key/value pairs of `AtString` and `AtParamValue`. (#8923)

## Incompatible changes

- **Plugin compatibility broken:** Arnold plugins such as shaders and procedurals will need to be recompiled to work in Arnold 6.0.0.0 and above.
- **GPU Driver requirements:** Arnold GPU and the Optix denoiser now require NVIDIA drivers version 435 or more.
- **Render callback signature change:** The render callback signature has changed, and it no longer receives the render status because that can be determined directly from the update type. The display outputs hint has moved to the `AtRenderUpdateInfo` argument which has been added, and which also contains other useful pass and sample setting information. (#8818)
- **AiRenderSetInteractiveOutput(uint32\_t)/AiRenderGetInteractiveOutput() deprecated:** Interactive output management should be done with the new `AiRenderAddInteractiveOutput(uint32_t)` and friends. The old set/get methods still work for the time being, however. (#8818)
- **Remapped Oren-Nayar diffuse roughness:** Scaling the `standard_surface.diffuse_roughness` and `car_paint.base_roughness` by a factor of 4.44 will restore the previous look. (#8839)
- **AiMicrofacetBSDFAverageFresnel()** has been replaced by the more general `AiBSDFAlbedo()`. (#7479)
- **Albedo AOV changes:** following a more principled definition of "albedo" `sheen_albedo`, `coat_albedo` and `specular_albedo` (in some cases) will return a black result. (#7479)
- **enable\_dithered\_sampling and enable\_progressive\_pattern:** The deprecated global options `enable_dithered_sampling` and `enable_progressive_pattern` have now been removed. Dithered sampling is now always used and a progressive pattern is always used when rendering in progressive mode. (#8555)
- **ARNOLD\_LICENSE\_MANAGER environment variable deprecated:** Licensing priorities can be now configured through the new environment variable `ARNOLD_LICENSE_ORDER` or the standalone tool *Arnold Licensing Manager*.
- **Autodesk Network Licensing version:** Autodesk Network Licensing requires now new license files with an updated 2020 version.
- **C++11 API:** We have raised our minimum C++ requirement to C++11. Compiling with C++03 will likely result in compilation errors. (#8729)
- **macOS minimum version is now 10.11:** We have raised the minimum version required for macOS from 10.9 to 10.11. (#9080)

## Bug fixes

### 6.0.0.0

#8789 `AiNodeResetParameter` not resetting links

#8728 [Alembic] Alembic procedural overwrites render callback and possible data race

#8999 [Alembic] Remove warning for scalar non-array properties using first entry

#8982 [Alembic] Unable to override points mode

#6956 Color Manager: Use well known OCIO config defaults to compute chromaticities

#7479 Common BSDF albedo API and more accurate albedo AOVs

#9034 Crash on rendering alembic with void or invalid filename

#8972 Crash when a procedural with externally added nodes is deleted

#8913 Crash when calling `AiRenderInterrupt` with a closest filter

#8995 Crash with include\_graph operator when changing target

#8821 Data race in AiRenderSetInteractiveOutput

#8525 Error in maketx when generating DWAA from single channel images

#8800 Follow the Optix 6.5 convention for the cache path on Linux

#8705 Get full AOV list even when not using progressive\_show\_all\_outputs

#8894 [GPU] AiNodeSetDisabled not working in IPR

#8778 [GPU] atmosphere\_volume.density and standard\_surface.transmission\_depth crash

#8283 [GPU] Background's alpha not handled correctly

#8980 [GPU] Banding artifact in volumes in progressive mode

#8819 [GPU] Cache pre-population can use too much memory

#8934 [GPU] Crash on destroying the mesh\_light's mesh in IPR

#8373 [GPU] Crash on interrupting alembic subdivision

#8988 [GPU] Crash on resizing volumes in Maya while in IPR

#9047 [GPU] Crash when volume shader is NULL

#8909 [GPU] Empty image filename doesn't produce a missing texture color

#8643 [GPU] Exceptions thrown during UpdateNodeGPU cause a crash

#7608 [GPU] Handle rays starting in overlapping volumes

#8823 [GPU] Image node crashes in scenes with textures disabled

#9016 [GPU] Incorrect skydome color with constant color inputs

#8879 [GPU] Incorrect Z AOV

#9076 [GPU] Render progress percentage can exceed 100%

#8716 Inconsistent return values from AiRendering() and AiRenderGetStatus()

#7525 Instanced volumes cause crash when volume\_update returns true

#8856 IPR sample degradation should upgrade in reverse order

#8792 maketx infinity checks are always done with 16-bit floats

#9026 MaterialX: Omit default shader material assignments

#8989 Operators not cooked properly in IPR with Render API

#6431 [OSL] Add standard include directories to include path

#8874 [OSL] oslc does not find stdosl.h

#8237 Progressive render consumes a lot of memory

#8466 Python bindings for AtRenderUpdateCallback crash on windows

#8933 Python render callback crashes on Windows

#8916 Regression in per-camera AOVs with a closest filter

#8938 Render callback should get AI\_DISPLAY\_OUTPUT\_PARTIAL\_INTERACTIVE on interrupt

## 6.0.1.0

#9172 [Alembic] alembic\_proc.dylib doesn't load on macOS < 10.14

#9167 [Alembic] Crash with NULL shader applied to Alembic procedural

#9132 Cannot destroy nodes at AI\_RENDER\_UPDATE\_BEFORE\_PASS time

#9100 Crash after destroying a camera-visible light

#9194 Crash in AiASSWrite() when a node has a NULL matrix array

#8955 Crash when loading incomplete camera definition

#8952 Crash when loading malformed polymesh nodes

#8954 Crash when loading quad lights with missing vertices

#9007 Deleting upstream operator nodes is not reflected correctly in IPR

#9138 Double check out of RLM and CLM licenses on same machine

#9102 Error in maketx when using DWAA or DWAB compressed EXRs from single channel images

#9008 Error when exporting to .ass after deleting an operator from a graph

#9084 Fast progressive accumulation filters do not work with negative AA

#9151 [GPU] Changing skydome exposure in IPR can trigger a recompilation of all GPU programs

#9112 [GPU] Crash after insufficient device memory error

#8817 [GPU] Crash when toggling "Debug Shading" options during IPR

#9065 [GPU] Crash with empty VDB volumes

#8619 [GPU] Crash with motion blur on points in sphere mode

#9115 [GPU] Error on changing the volume filename during IPR

#9145 [GPU] Manage parameter type conversion of shaders connected to cameras

#9136 [GPU] NaN values from Color Correct node

#9171 [GPU] Options node missing gpu\_support metadata

#9018 [GPU] Overblurred textures in shading networks with multiple image nodes

#8231 [GPU] Peak memory used not accurate on multi-GPU systems

#9071 [GPU] Regression causing subdivision and displacement to be single threaded

#9165 [GPU] Shape nodes are missing gpu\_support metadata

#9085 [GPU] Transmission scatter does not work if the standard surface is not directly connected to the shape

#9162 In progressive render, the AA 1 pass is not always skipped

#9096 [License Manager] License Manager appears underneath the menu in DCCs

#9015 [License Manager] Manager does not handle 'none' license type

#9129 [License Manager] 'SDK was already shut down' error when clicking on Sign in

#8873 [MaterialX] No node definition found error when connecting closure shaders

#9009 Oriented curves artifact when scaled

#9149 Python 3 incompatible print statement usage in ai\_device.py

#7888 Subdivision meshes sometimes disappear during IPR

#8979 Updated nodes are not systematically reprocessed by operators in IPR