

HtoA 5.6.0.0

28 Jan 2021

HtoA 5.6.0.0 is a feature release bringing light mixing, bloom, and interactive denoising via imagers and improved USD support. It features [Arnold 6.2.0.0](#) and [Arnold-USD \(#244e5d73, USD v20.11\)](#).

Installation

1. Download [Arnold for Houdini](#)
2. Follow these [installation instructions](#)

System Requirements

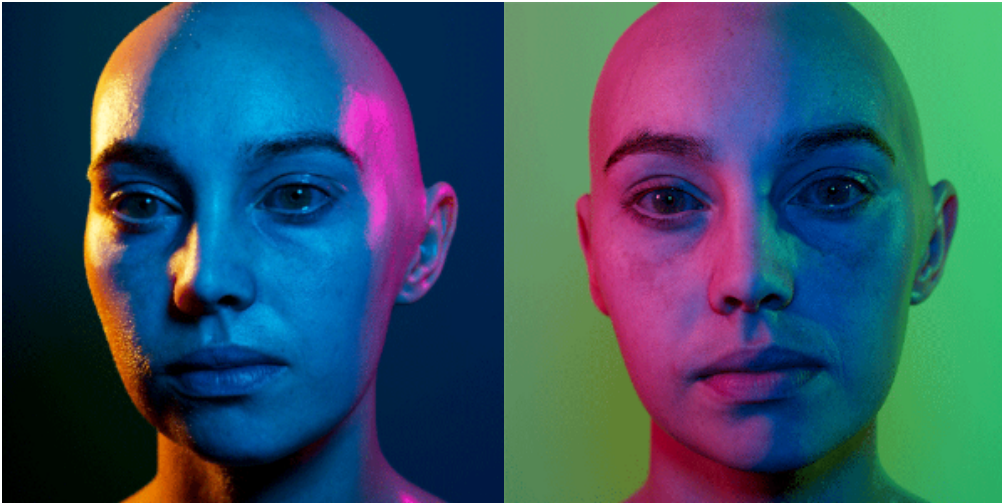
- Houdini, Houdini FX, Houdini Indie and Houdini Education
 - 17.5.460
 - 18.0.597
 - 18.5.462
- Windows 10 or later, with the Visual Studio 2019 redistributable.
- Linux with at least glibc 2.17 and libstdc++ 4.8.5 (gcc 4.8.5). This is equivalent to RHEL/CentOS 7.
- macOS 10.13 to 10.15. Note that macOS 11 Big Sur is not certified.
- CPUs need to support the SSE4.1 instruction set.
- GPU rendering works on Windows and Linux only and requires an NVIDIA GPU of the Ampere, Turing, Volta, Pascal, or Maxwell architecture. We recommend using the [460.39](#) or higher drivers on Linux and [461.40 \(Quadro\)](#), [461.40 \(GeForce\)](#), or higher on Windows. See [Getting Started with Arnold GPU](#) for more information.
- Optix™ denoiser requires an NVidia GPU with [CUDA™ Compute Capability 5.0](#) and above.

Enhancements

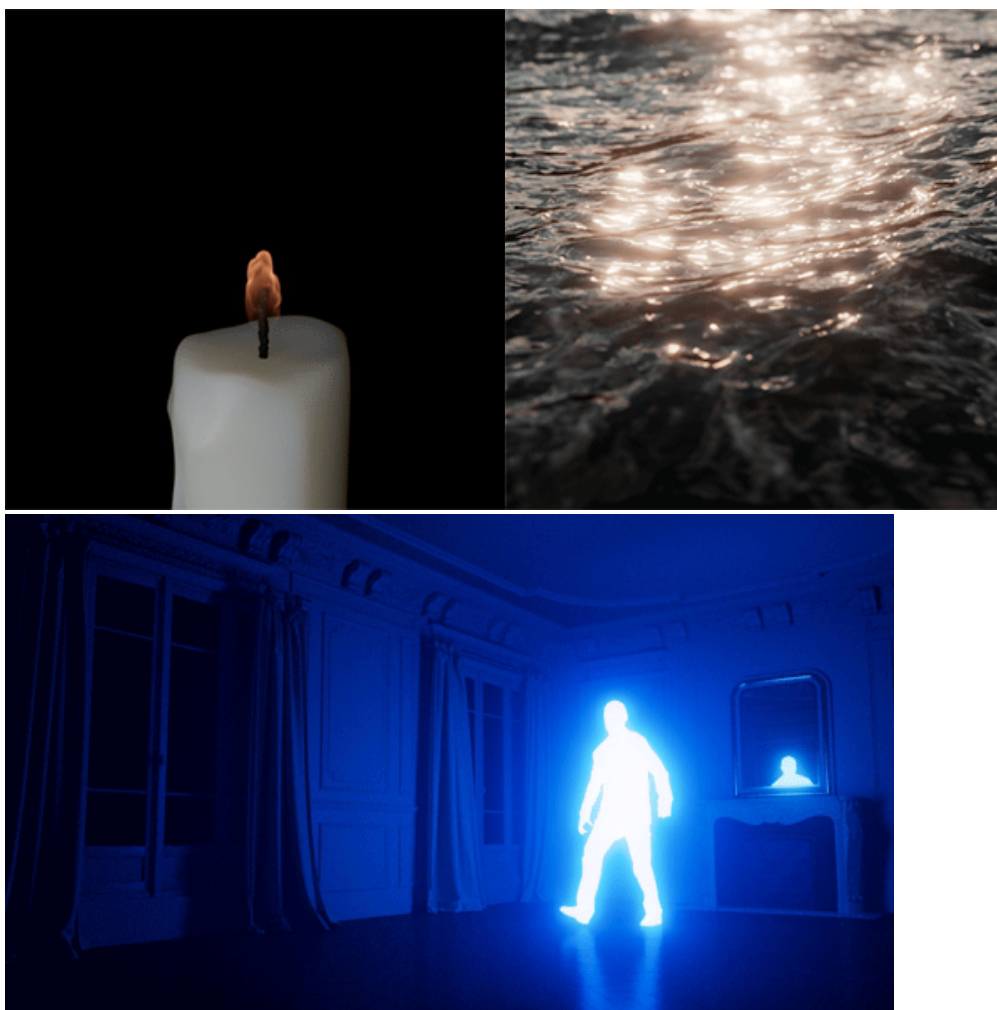
Imagers:

- `imager_denoiser_noice`
- `imager_denoiser_optix`
- `imager_lens_fx (bloom)`
- `imager_light_mixer`

Light Mixer imager: This new imager makes it possible to interactively edit the contribution of light group AOVs during and after rendering without restarting the render (core#9838).



Bloom in `imager_lens_effect`: The `imager_lens_effect` now implements a bloom effect. Bloom or glow is a post-processing effect that will blur pixels above a given threshold across the frame to simulate light bleeding on an imperfect lens. See the [\(core#9728\)](#).



- **Noise denoiser imager:** The **Arnold Noise denoiser** is now also available as a post-processing effect. You can automatically denoise images every time you render a scene, edit the denoising settings and see the resulting image directly in the render view. Note that `imager_denoiser_noise` does not support temporal denoising (core#9718).
- **OptiX denoiser imager:** The OptiX™ denoiser is now available as a post-processing effect. The imager also exposes additional controls for clamping and blending the result. (core#9719).
- **OpenColorIO v2 support:** Arnold now uses an updated version of **OCIO** and is able to load and run **OpenColorIO v2** configurations. It is fully backward compatible and will load and run existing configurations as before.(core#7827)
- **Up to 256 light AOVs supported:** The number of supported **light AOVs** is now 256, it was 15 previously (core#7802).
- **Faster cell_noise:** The `cell_noise` shader is now about 2.5x faster on CPU and 4.3x faster on GPU. In order to achieve this, we had to make a look-breaking change that manifests as a different random seed (core#9866).
- **New default values for Standard Surface:** The `base` and `base_color` parameters on `standard_surface` are now respectively set to 1.0, and (0.8, 0.8, 0.8) by default. Changing `base` to 1 after connecting `base_color` to a texture is such a common use case that it's better to swap the two defaults and avoid having to adjust `base` every time (core#9128 [standardsurface#15](#))
- **Improved progressive sampling:** faster and higher-quality sampling improves progressive and **adaptive** rendering performance (core#10023).
- **Constant color detection in maketx:** Added constant color detection to `maketx`. `maketx/AiMakeTx` will by default add the `--monochrome-detect` flag (core#10179).
- **OpenEXR maketx passthrough:** OpenEXR file inputs to `maketx/AiMakeTx` will now by default generate an OpenEXR based `.tx` files instead of TIFF based `.tx` files. `--format tif` or specifying a non-tx extension in the output filename will still allow for using other filetypes (core#6792).
- **maketx allows by default half TIFFs:** `maketx/AiMakeTx` will now by default allow creating TIFF based `.tx` files with half floats. Before, passing `-d half` would by default produce a 32-bit float TIFF `.tx` file. Beware that most tools do not support half TIFF, so float TIFF or half OpenEXR should be used when the `.tx` file needs to be used outside of Arnold (core#6792).
- **Faster maketx half TIFF generation:** Generating half TIFF `.tx` files should be several times faster on certain Linux machines (core#10224).

- **Support for OSL shader metadata:** Metadata from [OSL shaders](#) are now visible on Arnold nodes and can be read via the [AiMetadata](#) APIs (core#6718).
 - **Limit number of per-light log messages:** If there are more than 100 lights, we stop outputting per-light sampling log messages at the regular log verbosity and instead only output these at debug level verbosity. This should help avoid "spamming" the logs when there are many lights in a scene (core#10098).
 - **Faster Autodesk Analytics Program:** ADPClientService should upload data more quickly and consume fewer CPU resources, especially when there is no internet access (core#9776).
 - **Updated single-user licensing:** Autodesk Desktop Licensing has been updated to version 11.0.0.4854. This version is not compatible with previous Arnold versions and should be installed automatically from plugin installers if single-user licensing is used, or can be updated manually from the [Arnold License Manager](#). It is required only for single-user licensing, network licensing (AdskFlex) and RLM are not affected (core#289).
 - **More accurate crash report:** When crashes occur while Arnold is running, a stacktrace is printed out. The stacktrace report should now be more reliable and slightly easier to understand (core#9156).
 - **More profile blocks added:** Most of the Arnold API now has associated [profiling blocks](#) so it is easier to determine which Arnold components are consuming the most render time (core#10272).
 - **Interactive imagers in kick:** kick now displays imagers in interactive mode (core#9836).
 - **Faster Apple M1 performance:** Arnold should now be able to properly use both the big and small M1 cores (still under Rosetta2) (core#10062).
 - **OpenVDB 7.1:** Arnold now uses OpenVDB 7.1. More information about the changes between OpenVDB 4 and 7.1 can be found [here](#). (core#9861)
-
- **Imager support:** including using the `bypass` flag as enable parameter (htoa#1538, htoa#1615)
 - **Imager IPR:** IPR support for imager creation, parameter changes, linking, and deletion (htoa#1633, htoa#1640, htoa#1651)
 - **Light Mixer Setup:** Populate and update `light_mixer` imager with light groups in the scene (htoa#1621)
 - **Multiparm support:** multiparms can be defined via metadata for custom nodes, and this is used for the `set_parameter` ROP and `light_mixer` imager (htoa#1596, htoa#1597, htoa#1599)
 - **Array defaults:** Adding `houdini.default` metadata to support overrides of parameter array defaults (htoa#1601)
 - **Optix denoiser:** Support the `optix` denoiser via imagers without the `optix_filter` (htoa#1606)
 - **USD material overrides:** Support material overrides using the `shop_materialpath` attribute for packed usd primitives (htoa#1644)
 - **Render settings LOP updates:** Updates to `RenderSettings` LOP adding search paths and `RenderGeometrySettings` LOP with `autobump` visibility ray settings (htoa#1611, htoa#1609)

Bug Fixes

- htoa#761 Fix sticking first `ramp` key, and correct the parameter disable logic
 - htoa#982 Fix `forcelights` parameter on the Arnold ROP
 - htoa#1521 Per light AOV's display incorrectly in render view on Linux
 - htoa#1582 Allow optional `gpu_sparse_textures` parameter
 - htoa#1608 Remove subsurface ray visibility from `RenderGeometrySettings` LOP
 - htoa#1618 Fixed crash rendering `openvdb` files after `openvdb` 7.0 upgrade
 - htoa#1630 Fix a crash updating `ramp` shaders in IPR
 - htoa#1632 Fix IPR crash on node deletion
 - htoa#1636 Skip bad color spaces referenced in `ocio` color space families
 - htoa#1641 Fix to allow installer silent mode
 - htoa#1642 Create `usd` procedurals for all `usdType` intrinsic packed primitive types
 - htoa#1643 USD packed primitives passthrough for inactive objects prevent
 - htoa#1648 Light mixer per element array update during IPR
 - htoa#1649 Fix override for usd packed primitives
-
- core#10278 100% progress is not always printed.
 - core#10074 Blackbody: bad XYZ color normalization.
 - core#9982 Clamp negative values with `thin_film` and `transmission`.
 - core#10233 `clip_geo`: stops working if you tumble the camera in IPR.
 - core#9744 Crash due to out-of-memory when rendering.
 - core#10222 [GPU] Fix Arnold not being able to recover from recoverable exceptions.
 - core#10208 [GPU] Remove usage of scientific notation for log progress.
 - core#10175 Mix of two `standard_surfaces` with `transmission` hangs the render.
 - core#10149 Points should be visible in the viewport API in polygon mode.
 - core#10091 `quad_light` can produce rare black samples.
 - core#10283 Remove MacOS `xcrun` installation prompt when crash occurs.
 - core#10259 Report error code when NVML fails to initialize.
 - [usd#592](#) Invalid face-varying primvars crash the render delegate.
 - [usd#596](#) Invalid USD is produced if polymesh is made of triangles and `nsides` is empty.
 - [usd#481](#) `std::string`, `TfToken`, and `SdfAssetPath` typed `VtArrays` are not converted when setting primvars.
 - [usd#619](#) Several built-in render buffer types are not translated to the right Arnold AOV type.
 - [usd#634](#) Fixing disappearing meshes when playing back animation.
 - [usd#621](#) UVs not read from facevarying primvar if indexes are not present.
 - [usd#638](#) Motion start and motion end is not set reading animated transformation.