

KtoA 2.3.0.0

Release Date

March 20, 2019

This release includes the [beta version of Arnold GPU](#)



Important information about Arnold GPU (beta)

- Check the [system requirements](#) before using Arnold GPU. If you don't have a [supported card](#) or the [required drivers](#), GPU rendering will not work.
- Review the list of [supported features and known limitations](#) before you start using Arnold GPU.
- If you have any technical problems, questions, or feedback on Arnold GPU, use the beta forum on [Arnold Answers](#)

Download and Installation

Arnold, KtoA, and other downloads are available [here](#). Installation instructions come with KtoA, but can also be viewed here: [Installation](#).

Compatibility

- **Arnold:** 5.3.0
- **Katana:** 3.0v1+, 3.1v1+
- **Platforms:**
 - Linux: x86-64, RHEL 6+ or compatible glibc
 - Windows: 7+ on x86-64, with VC++ 2015 redistributable installed
- **GPU (beta):** see [here](#) for detailed information. Required NVIDIA drivers:
 - **Linux:** 418.56 or higher
 - **Windows:** 419.67 or higher

Enhancements

- **Update to Arnold 5.3:** Updated to Arnold 5.3, with many new features and bug fixes. See the [release notes](#) for more information. A few of the highlights are below. (#365)
 - **GPU rendering (beta):** While not intended yet for production, and not supporting all CPU rendering features, you can switch the device in `ArnoldGlobalSettings` to GPU to try out rendering on Nvidia-based GPUs. Note that if you start out with GPU rendering, you can switch back and forth between CPU and GPU rendering during live rendering to do some comparisons. Please see the core release notes (included in KtoA) for more information. (#252)
 - **Visible lights:** `point`, `quad`, `cylinder` and `disk` lights all now have camera and transmission parameters which allow them to be visible in camera and refraction, respectively. They are zero by default to match legacy behavior, but when turned on activate a visible lights workflow so you no longer have to create proxy geometry with emission to make them visible. (#368)
 - **Shader additions:** Ramp shaders gained a new time mode, `standard_surface` gained anisotropy controls for the coat layer and `transmission_extra_roughness` now accepts negative values to dial back roughness compared to the specular layer, and an enhanced version of the randomwalk SSS engine has been added.
 - **Improved adaptive sampling:** For the adaptive engine to perform well, progressive rendering no longer needs to be used. The convergence metric has improved substantially, and the threshold value is much easier to understand and control. With this change, it should be reasonable to use adaptive sampling most of the time.
 - **Smart opaque:** built-in shaders now can set an object's `opaque` flag automatically for most cases. A few limitations remain, so you may need to still set `opaque` off manually for curves and points with `min_pixel_width` and when using OSL shaders.
- **Show all outputs during blocky progressive rendering:** `ArnoldGlobalSettings` gained an option to force all outputs to be shown during blocky progressive passes for live rendering. Previously, only the first or primary output would be shown for the faster blocky progressive passes to improve speed of interactivity, but for longer renders it is desirable to show all outputs right from the start to view all AOV data in the monitor right away. (#365)
- **Operators on procedurals:** Renderer procedural locations now use an `arnoldOperator` shader slot to attach the operator network to the specific procedural. Just insert the operator into your material and assign it to your renderer procedural location. (#363)
- **Include operator:** The `include_graph` operator is now available to include an external `.ass` file into your scene, including any operators. If you use an `include_graph` operator on a renderer procedural location, it will include that `.ass` file contents into the procedural's namespace, rather than the global namespace. (#362)

Incompatible Changes

- **Deprecated shaders hidden by default:** Deprecated shaders, such as the old `standard` shader, are now hidden from the list of shaders by default. To force them to show up again, set the environment variable `KTOA_ENABLE_DEPRECATED_SHADERS` to any value. (#372)
- **Shaders specific to other DCCs hidden by default:** A few shaders that are used in C4D and Maya, but which are not intended for cross-DCC workflows, are hidden by default from the shader list, although they are present with Arnold core now. To enable them in the list, only if you must, set the environment variable `KTOA_ENABLE_OTHER_DCC_SHADERS` to any value. (#365)

- **Renamed global options:** `adaptive_threshold` is renamed to `AA_adaptive_threshold`, please use the update script to update your scene. (#365)
- **New defaults for adaptive sampling:** Adaptive sampling has been vastly improved, but along with the improved engine new defaults were needed for `AA_samples_max` and `AA_adaptive_threshold`. It is recommended to use the new default threshold to start, and then adjust from there. (#365)
- **standard_surface defaults changed:** The `specular_ior` default changed from 1.52 to 1.5, and the `specular_roughness` default changed from 0.1 to 0.2. (#365)
- **NVIDIA Kepler GPUs no longer supported:** For the optix denoiser these older GPUs are no longer supported; only GPUs with CUDA compute capability 5.0 on up are used. (#365)
- **Remove support for Katana 2.x series:** Only Katana 3.0 and 3.1 series are supported now.