

KtoA 2.0.0.0

Download and Installation

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

Important Note

This is a major release of KtoA based on the new Arnold 5.0 that brings with it numerous incompatibilities with previous (KtoA 1.1.x) releases. Many incompatibilities are resolved by simply running the `UpdateArnoldNodes` shelf script after loading previous katana files.

Of note, to bring your assets and lookdev in line with Arnold 5 you will need to make changes to your shading networks. Most older shaders will continue to work, but you should expect visual differences. It is highly recommended to move to the new standard set of shaders, and check existing shader connections and parameters. Also, for objects you should check ray types such as visibility, double-sidedness, sample counts and ray depths.

Compatibility

- **Arnold:** 5.0.0+
- **Katana:** 2.5v4+
- **Platforms:**
 - Linux: x86-64, RHEL 6+ or compatible glibc
 - Windows: 7+ on x86-64, with VC++ 2015 redistributable installed

Enhancements

- **Update to Arnold 5.0.0.0:** KtoA is now updated to and requires Arnold 5.0.0.0, bringing many changes and enhancements. See [Arnold 5 release notes](#) for more information. (#124, #128, #131)
- **New standard shaders:** The old `standard` and `hair` surface shaders have been deprecated in favor of new layer-based shaders `standard_surface` and `standard_hair`. Also, the `volume_collector` shader from the common shaders is deprecated and supplanted by `standard_volume`. The old shaders will still work, but it is highly recommended that assets be ported to use the new shaders. They bring many benefits: proper Fresnel handling, energy conservation, per-layer normals, and are more physically-based right out of the box. The parameters are quite different from the old shaders, but are quite intuitive for those who are familiar with layer-based shading systems such as `aiShaders`. Please see the following areas of the Arnold documentation for the new standard shaders: (#124)
 - [standard_surface](#)
 - [standard_hair](#)
 - [standard_volume](#)
- **New common shaders:** The common shaders are now built in to Arnold itself; in addition, new shaders have been added: `curvature`, `flake`, `camera_projection`, `complex_ior`, `two_sided`, `normal_map` and `vector_map`. (#129)
- **Update OpenVDB plugin:** The OpenVDB volume/implicit plugin has been updated for Arnold 5, and is now built-in to Arnold itself. (#130)
- **Build system updated:** For developers, the build system is now unified on both Linux and Windows. This shouldn't have any effect on end users. (#135)
- **OSL and closure shaders:** Open Shading Language, along with C++ closure-based shaders are now the main shading mechanism in Arnold. Note that `.osl` or compiled `.oso` shaders must be placed in the plugin searchpath in order to be found. Arnold will attempt to compile the shaders to `.oso` if only `.osl` shaders are found. Please see the [documentation](#) for more information. (#125)
- **Light path expressions:** Previously a holdover from the SPI days of Katana, there was an array parameter `lightExpressions` on the `ArnoldOutputChannelDefine` node which did nothing in Solid Angle's version of Arnold. That has been removed, and instead a single `lightPathExpression` parameter has been added. When it is filled out, shaders do not need to register AOVs and instead closure shaders will automatically detect when the light path expression has been met and will write to the output. Also, `lightGroups` has some additional controls on which lights groups are included in the output, and whether they are written out to separate AOVs in an EXR or TIFF file. A custom light group may be set as well. Please see the [documentation](#) on LPEs for more information. (#145)
- **Visible lights:** Skydome lights are now visible by default in camera rays, and also perfect reflection and refraction rays. Note that skydome lights will also occupy the background if the `camera` parameter on any skydome light is set to greater than zero. (#124, #128)
- **Arnold 5 update shelf script:** The update shelf script has been extended to do some of the legwork of updating projects from KtoA 1.1 / Arnold 4.2 to KtoA 2.0 and Arnold 5.0. It also will allow undoing its effects properly after it runs, and it won't deselect any selected nodes in the node graph. (#132)
- **Per-object motion blur customization:** Using `ArnoldObjectSettings` objects can have their motion range, number of transform samples, and/or number of deformation samples customized from the global motion samples set in `RenderSettings`. (#143)
- **OpenVDB improvements:** The `ArnoldOpenVDBSurface` and `ArnoldOpenVDBVolume` nodes are much improved; they automatically provide the bounds of any data in the viewer, and they have transform parameters that are interactively editable with the viewer manipulators. They have improved defaults, help hints, and UI management of step parameters. (#48)
- **Yeti improvements:** Yeti is now given frame-relative motion sample times (per Arnold 5). It will try to display the bounding box of the Yeti cache if the `pgYetiCacheInfo` executable is in the PATH. Users can now also manipulate the transform of the Yeti location in the viewer. (#154, #106)
- **Deferred expansion removed:** Procedurals, volumes and implicits now all load at startup and are immediately expanded in Arnold. This may cause a slightly higher startup time, but generally results in faster renders overall. This is equivalent to setting `proceduralExpansion` to `at_startup`. That option has been removed, and any consideration of `forceExpand` attributes on locations also has been removed. When writing to `.ass` files, however, a new parameter `openProcedurals` on the `ArnoldGlobalSettings` node has been added to control whether procedurals are expanded in the resulting exported `.ass` file. (#153)

- **Bounds slack for all volume plugin nodes:** For all volumes that are backed by a plugin, such as OpenVDB volumes, the `bounds_slack` parameter in `ArnoldObjectSettings` can be set. This is a standard convention for volume plugins in Arnold, giving space for shaders to add positional variance to add resolution to volumes beyond what is stored on disk in a grid. (#153)
- **Multi-threaded scene translation:** With the removal of deferred procedural expansion, previously parts of the scene would be translated in parallel inside of the katana procedural, but now it is all taken care of up front. Scene translation is now multi-threaded, where in many cases locations can be processed in parallel. A new control, `translationThreads` in the `ArnoldGlobalSettings` node has been added to customize this. Note that in many cases, having a high number of translation threads is counterproductive; the appropriate range seems to be between 4-12 threads, depending on scene complexity. (#155)
- **Randomized child order:** Particularly in batch rendering, when many frames start on a render farm at the same time, network file servers can become congested if the same file resources (such as Alembic files) are requested all at the same time. Randomizing the order in which locations are processed (by frame) may help alleviate this bottleneck. This is not active by default, as it does introduce a small amount of overhead, but may help with some network and NAS setups. (#155)
- **Pref on pointcloud locations:** The attribute `geometry.point.Pref` is now recognized and translated as varying user data on `pointcloud` locations. (#139)
- **Deprecation warnings for shaders:** Shaders that are marked *deprecated* will now have a `__warning__` parameter added with a message about the shader being deprecated, so that it's obvious which are slated for future removal. Currently, the set of deprecated shaders are: `hair` (use `standard_hair` instead), `skin` (use `standard_surface`), `standard` (use `standard_surface`), and `volume_collector` (use `standard_volume`). (#162)
- **Host information in logs and ass files:** `host app` information is now added to logs and exported ass files indicating the KtoA version, OS, katana build version, and katana runtime version. (#165)

Incompatible Changes

- **Removed arnoldBump shader slot:** The new standard shaders take a normal vector as a parameter for each layer, so it is no longer needed nor advisable to modify the global shading normal in a shader network, but instead plug in the modified normal directly to each corresponding shader parameter. As such, the bump shader slot has been removed in Arnold 5, and any shader or shader networks that were connected to this slot in KtoA will now be ignored. These will need to be moved into the regular surface shading network instead. (#142)
- **Auto step size as default for OpenVDB:** For OpenVDB volumes and surfaces, a new `auto_step_size` parameter has been added to make it obvious when using the natural step size from grid data. If you previously customized the step size, you will need to turn off `auto_step_size` to restore your previous customization. It is usually preferred to use the auto step size and instead modify `step_scale` to go relatively higher or lower. (#48)
- **Removed unneeded filter types:** Various infrequently-used and unneeded pixel filter types have been removed from `ArnoldGlobalSettings`. If you used one of these, it will revert back to the gaussian filter. (#128)
- **Plugins as node entries:** Procedurals, volumes and implicit Arnold nodes. KtoA helper Katana nodes for creating them have been updated, but for your own procedurals you will need to set `rendererProcedural.node` (string) to be the name of the node type your plugin provides. You can still use the `rendererProcedural.dso` path to have KtoA load your plugin, but you can also put it in the procedural searchpath or the `ARNOLD_PLUGIN_PATH` environment variable. (#126, #127)
- **New motion blur specification:** Internally, KtoA has always used frame-relative motion samples and send them to Arnold as such (except when `compatible_motion_blur` was turned on). Arnold itself has removed `transform_time_samples` and `deform_time_samples` in favor of uniformly distributing motion samples between `motion_start` and `motion_end` float parameters, and KtoA now honors this. If there are non-uniform motion samples in Katana, KtoA will sample uniformly from them; this means if you have complex motion that needs particular attention, you may need to increase the number of motion keys to capture it. (#143)
- **OpenVDB node parameter changes:** The `ArnoldOpenVDBSurface` and `ArnoldOpenVDBVolume` nodes have some parameter changes; please run the update shelf script to get the new parameter names. (#130)
- **Increased default ray depths:** The default ray depths are now one each for diffuse and specular bounces, eight for transmission bounces, but still zero for volume indirect. If you had left those values at the defaults in your katana recipes, you may now have some indirect diffuse and specular light show up where you didn't have it before, with a corresponding increase in render time. If this is incorrect for your scene, go set them lower in the `ArnoldGlobalSettings` node. (#124)
- **Ray types:** For background visibility, object visibility, autobump visibility, and sidededness a few ray types have changed. `AI_RAY_DIFFUSE` has been changed to `AI_RAY_DIFFUSE_REFLECT`. `AI_RAY_GLOSSY` has been changed to `AI_RAY_SPECULAR_REFLECT` and `AI_RAY_REFLECTED` has been subsumed into it as well. `AI_RAY_REFRACTED` has been changed to `AI_RAY_SPECULAR_TRANSMIT`, and `AI_RAY_DIFFUSE_TRANSMIT` has been added to differentiate rays that go below the surface normal but are too diffuse to be considered glossy/specular. The update script honors these interpretations, but you should consider double-checking your object settings for any you have customized to make sure they still do what you expect. (#124, #132)
- **Additional changed and removed parameters:** (#128)
 - `subdiv_dicing_camera` has been moved from per-polymesh settings to be a global option
 - `auto_transparency_threshold`, `auto_transparency_mode` and `enable_fast_opacity` have been removed as they are no longer needed
 - `ignore_direct_lighting` has been removed as the AOV system can be easily used to achieve similar output
 - `bump_space` has been removed as bump is always done in object space now
 - `enable_displacement_derivs` has been removed, and displacement derivatives are now always calculated
 - `GI_glossy_samples` and `GI_glossy_depth` have been renamed to `GI_specular_samples` and `GI_specular_depth`, respectively, and include perfect specular and glossy reflections.
 - `indirect_glossy_blur` has been renamed to `indirect_specular_blur`.
 - `GI_refraction_samples` and `GI_refraction_depth` have been renamed to `GI_transmission_samples` and `GI_transmission_depth` respectively, and include all forms of below-surface rays.
 - `GI_reflection_depth` has been removed; this is subsumed into `GI_specular_depth` that handles both perfect and glossy specular reflections
 - `GI_single_scatter_samples` has been removed; the other forms of subsurface scattering should be used instead of the old single-scatter mode
 - `GI_falloff_start_dist` and `GI_falloff_end_dist` have been removed in favor of physical falloff always being on. The decay light filter can still be used to customize falloff.

- `light_gamma`, `shader_gamma` and `texture_gamma` have been removed, as it is now expected that all inputs are color-corrected to be stored linearly now (as has been true for Katana for a long time). Textures may have their input colorspace specified via the `color_space` parameter on the `image` shader node. Arnold 5 itself also supports custom `color_manager` nodes for even more color space management options.
- `bucket_scanning` has had rarely-used `bottom`, `right`, and `woven` modes removed
- `texture_glossy_blur` has been renamed to `texture_specular_blur`
- `shader_searchpath` has been renamed to `plugin_searchpath`

Bug Fixes

- #152 Pixel aspect ratio set incorrectly
- #148 Intermittent crash using `ArnoldOutputChannelDefine`
- #151 Viewed (blue) node fails update script if it gets changed
- #157 Use `render interrupt` instead of `abort` when pausing a render
- #158 texture data on instances uses wrong number of polygons