6.0.3.0

20 Apr 2020

Enhancements

- **New instancer procedural**: A new `instancer` procedural node has been added to create instances of shapes and lights. (#7901)
  The nodes to instance are defined by the `nodes` and optionally `node_idxs` parameters. Per instance transformation, visibility and shaders can be specified via the `instance_matrix`, `instance_visibility` and `instance_shader` parameters. Any other per instance parameter (builtin or user parameters) can be specified via array type user parameters prefixed as `instance_`, for example:

  ```
  declare instance_intensity constant ARRAY FLOAT
  instance_intensity 1.0
  ```

- **Parallel execution of driver functions**: More driver functions can now be run in parallel. This reduces the overhead for running driver functions which slightly increases interactivity for fast renders. (#9268)
- **User defined OSL include paths**: A new option `osl_includepath` has been added to allow additional include search paths to be provided to OSL when compiling shaders. This option is a single string that can contain multiple search paths separated by a colon (`:`) or a semi-colon (`;`). (#8978)
- **kick Alembic, OBJ, PLY files**: New scene formats are supported: Alembic, .obj and .ply. They can be loaded through the `AiSceneLoad` API, or given as input files to kick. (#9446, #9464)
- **kick -default_nodes**: A new parameter in kick can be used to automatically create lights and/or camera when they are missing from the scene. With that, simple geometry files like `.obj` and `.ply` can be rendered to get an idea of their general look. (#9429)
- **UV Camera surface shading**: Shaders connected to the newly added `post_bake` parameter on `uv_camera`, will be run after ray casting on the baking destination surface. This is useful to bake surface properties, or to post process baked AOVs based on destination surface properties. (#9549)
- **Upgrade USD to 19.11 in the usd procedural**: The usd procedural is now built against USD 19.11. (#9406)
- **OIIO Upgrade to 2.2.1**: We have upgraded our OIIO to roughly 2.2.1. (#9040)
- **Overriding alembic instances with operators**: A new parameter `use_instance_cache` enabled by default has been added to the alembic node to control the internal cache that shares instances between alembic nodes. This parameter can be disabled when using operators to override the procedural contents while`make_instance` is enabled. (#9299)

GPU enhancements

- **OSL Textures**: Initial support for textures has been added to the GPU OSL runtime. (#9066, #9479, #9480)
- **Reduced texture memory usage**: The GPU renderer now supports the `texture_conservative_lookups` option. We observed a 15% memory improvement in some scenes. (#9263)
- **Custom AOVs**: Custom AOVs are now supported. This includes support for all aov_write_* shaders. (#9322)
- **AOV read shaders**: AOV read shaders are now supported. (#9396)
- **ID AOVs in Standard shaders**: The ID AOV parameters are now supported in the StandardSurface and StandardHair shaders. (#9397)
- **Photometric lights**: Support for photometric lights has been added to the GPU renderer. (#9424)
- **Noise improvement on volume rendering**: The quality of GPU volume renders has been significantly improved, particularly in strongly-colored translucent media. (#7636)
- **Wireframe shading**: The wireframe shader and the wireframe color modes in the utility shader are now supported. (#7734)

USD enhancements

- **Using the new instancer procedural**: The procedural now uses the core shipped instancer procedural. (usd#256)
- **Mask:** The scene format plugin now supports a mask parameter, allowing selective export of Arnold nodes. (usd#274)
- **Computed Primvars:** Computed primvars are now supported, enabling previewing UsdSkel and Houdini crowds. (usd#265 and #267)
- **Prim Visibility**: The visibility token from UsdGeomImageable is now correctly inherited. (usd#218)
- **Closures**: Closure attributes are now written to the USD file. (usd#322)
- **Options:** The options node is now correctly translated. (usd#320)
- **Improved Hydra Delegate Rendering**: The Render Delegate is now using the Arnold Render API correctly, leading to better responsiveness. (usd#270)
- **Improved Hydra Delegate Render Buffers**: The Hydra Render Buffer support is now significantly improved, including improved performance. (usd#8)
- **Hydra Delegate 32-bit buffers**: The render delegate now outputs 32-bit float buffers, instead of dithered 8 bit whenever possible. (usd#9)
- **arnold:global: prefix**: Prefixing Render Settings with `arnold:global:` is now supported.
- **Shaping Parameters**: Shaping parameters on Sphere Lights are now supported. This includes Spot and IES parameters, excluding IES normalize and IES angle scale. (usd#314)
- **Barndoor Parameters**: Solaris Barndoor parameters are now roughly approximated using the barndoor filter. Note, Arnold does NOT match Karma. (usd#332)

API additions

- **ADP opt-in**: We have deprecated `AiPromptADPDialog()` which does not work on Linux and instead added API functions for creating your own ADP opt-in dialog boxes. See the documentation for how to use the functions `AiADPIntroWasShown()`, `AiADPSetIntroShown()`, `AiADPSetOptedIn()`, and `AiADPDialogStrings()`. (#9351)
- **New custom scene format plugins**: This is a new plugin type which enables implementing custom scene formats for loading and/or writing to file and direct rendering through kick. See the API documentation for a full description of the new plugin interface. (#9303)
- **Scene format information**: We added new API functions to traverse the list of supported file formats `AiSceneFormatIterator()`, and their corresponding file extensions `AiSceneFormatGetExtensionIterator()`. There are also new methods to query information and supported metadata for all formats `AiSceneFormatGetName()` `AiSceneFormatGetDescription()`, `AiSceneFormatSupportsReading()`, `AiSceneFormatSupportsWriting()`, `AiSceneFormatGetMetadataStore()`. See the API documentation for full description. (#9391)
- **AiUniverseAddDefaultNodes**: This is a new API method to create default camera/light nodes when the scene has none. The type of node which will be created is controlled via optional parameters, with `default_lights` and `default_camera` acting as toggles. The `light_type` and `light_mode` parameters can be used to select the type and arrangement of created lights. (#9429)
- **Guarantee synchronous AiRenderRestart/AiRenderResume**: When a call to `AiRenderRestart`/`AiRenderResume` returns, it is now guaranteed that an attempt has been made to restart/resume rendering. I.e. a subsequent call to `AiRenderGetStatus`/`AiRendering` is guaranteed to return the status of rendering *after* the restart/resume has begun. Note: restarting or resuming can fail or finish fast, so it is not guaranteed that there is a render in progress at that point. (#9307)
- **Parallel execution of driver functions**: The `driver_needs_bucket`, `driver_prepare_bucket` and `driver_write_bucket` functions can optionally be executed in parallel if the corresponding `parallel_driver_needs_bucket`, `parallel_driver_prepare_bucket`, `parallel_driver_write_bucket` boolean metadata are set to true on the driver. Driver nodes not implementing those metadata will behave as before. (#9268)
- **Procedural node local data**: Procedural nodes now support `AiNodeSetLocalData` and `AiNodeGetLocalData` to store and retrieve the user defined node local data. (#9386)
- **AiParamValueMapMerge**: New API method `AiParamValueMapMerge` which merges the contents of a given param value map into another. (#9280)
- **New array type render hint**: Array type render hints are now supported using the call `AiRenderSetHintArray`. Currently no render hints of this type are defined. (#9346)

Incompatible changes

- **kick -ADP**: Previously `kick -ADP` opened a dialog box in OSX and Windows and did nothing in Linux. Now on all three platforms it will let you change your Autodesk Analytics Program opt-in status using the console. No dialog boxes are opened up and instead the selection is made by typing into the console. (#9351)

Bug fixes

- 9417 AiColorManagerTransform() crashes when calling outside of a render session
- 8684 Continue processing assignment expressions if the user parameter already exists
- 9537 Crash in IPR when cloning a visible mesh light
- 9390 Crash on a null uv_camera's mesh
- 9546 Crash when AA samples > 1024 and abort_on_error is false
- 9395 Crash when attempting to append to completed deep exr
- 9526 Crash when destroying a secondary universe
- 9103 Difference when converting single channel textures to ACEScg
- 9512 Duplicated AOVs causing noisy RGBA
- 9490 [GPU] Crash on destroying ai_default_reflection_shader after rendering
- 9313 [GPU] Crash when disabling motion blur during IPR session
- 7878 [GPU] Discrepancy in volume rendering when scatter_color components are large
- 9482 [GPU] Occasional illegal address when using autobump
- 9357 Hang due to a combination of parallel and serial procedural initialization
- 9552 kick 'n' is unable to run optix denoiser
- 9501 [Licensing] CLM v2 InstallHelper hangs on Windows when 5 or more adsk products are installed
- 9006 [MaterialX] Crash when using 3rd party shaders in MaterialX
- 9428 [MaterialX] Simplify node graph names when exporting materials
- 9554 OptiX denoiser does not render negative AA passes in non-progressive render mode
- 9548 [OSL] Multiline define statements in OSL are ignored
- 9031 [OSL] regex in OSL does not always work
- 9565 Report shader name when texture lookup fails
- 9273 rgb_to_vector should use RGB as input type
- 9558 Run OptiX denoiser on all non-negative AA passes
- 9528 [USD] Fix crashes when writing upper case extensions
- usd#295 [USD] Fixed a crash when using the Arnold Viewport functions.
- usd#288 [USD] Fixed a crash when writing USD files with upper-case extensions.
- usd#298 [USD] Fixed a crash when instancer nodes had uninitialized node pointers.
- usd#4 [USD] Fixed a bug with aborted renders not marking the render pass as converged.
- usd#320 [USD] Fixed bugs with the string export functions.
- usd#342 [USD] Fixed a crash when writing pointer attributes.