Release Notes for Katana 2.0v5

Release Date

25 September 2015

System Requirements

- Katana 2.0v5 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

Supported Renderers

Pixar's RenderMan, Solid Angle's Arnold, and Chaos Group's V-Ray are each supported by plug-ins supplied directly by those companies.

For Pixar's RenderMan, please contact Pixar to get RenderMan Studio for Katana (also called RfK). You also need to install the relevant version of the RenderMan renderer (RenderMan Pro Server).

For Arnold, please contact Solid Angle to get KtoA. This includes both the Arnold renderer as well as the Katana plug-in.

For V-Ray, please contact Chaos Group to get V-Ray for Katana. This includes the V-Ray renderer as well as the Katana plug-in.

All queries regarding third-party plug-ins should be directed to the relevant provider.

The Katana installation includes legacy plug-ins for PRMan 17.0 and Arnold 4.2, which should be mainly considered as reference to show how a render plug-ins can be implemented. For support of the latest renderer versions and features it is recommended to use the plug-ins provided directly by the relevant vendor company.

Third-Party Dependencies

Katana 2.0v5 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

Feature Enhancements and Important Changes

- BUG ID 49089 - The Doxygen-powered API reference now includes documentation for many more plug-in APIs available in Katana.
- BUG ID 50437 - Katana sometimes crashed if a large number of file descriptors were opened.

NOTE: If you are using Geolib3's FnRender module, you are advised to recompile your plug-ins to benefit from these fixes.

- BUG ID 50740 - Help text has been added for the **overscan** parameter of RenderSettings nodes to explain how to use the four values.
- BUG ID 50788 - **Scene Graph** tab custom columns may now observe the existence of group attributes. Existing group attributes are indicated with text of the form <**Group with** [n] **children**>.

Bug Fixes

- BUG ID 27167 - The **Place Specular** manipulator placed the light along the surface normal, instead of along the reflection direction.
  This version of the manipulator has been moved to a new **Place Along Normal** manipulator, and the **Place Specular** manipulator now behaves as expected.

- BUG ID 38294 - When starting multiple Katana sessions in batch mode on the same render farm machine at the same time, an exception was raised due to a race condition in the creation of a font cache directory.
- BUG ID 40546 - The placement of objects through DollyConstraint nodes was incorrect. The algorithm has been rewritten to place a camera so that it exactly fits the size of the bounds of the objects it is constrained to.

  Existing Katana projects that use DollyConstraint nodes are likely to require modification to account for changes in the nodes' results.
- BUG ID 47937 - Assigning conditional visibility to **dynamicArray** widgets raised an exception.
- BUG ID 49693 - After picking an object in the **Monitor** tab using the Pixel Probe tool and pressing the **Select in Scene Graph** button in the tab's toolbar, the picked object was not displayed as selected in the **Viewer** tab.
- BUG ID 49894 - ConstraintResolve: When calculating the resolved transform for a constrained location, the resulting **xform** attribute was missing time samples in some cases, such as when the location didn't have an existing **xform** attribute set.
- BUG ID 50020 - When scrubbing numeric parameters in the **Parameters** tab of certain nodes, such as Material nodes, widgets of array parameters flickered.
- BUG ID 50021 - The **initialiseCaches()** method in the RendererInfoPlugin API is now deprecated. Do not rely on this method when writing renderer plug-ins: in Katana 2.1 onwards, it is no longer called.
- BUG ID 50034 - CEL: Evaluating a CEL expression that referenced a collection that made use of binary set operators, including the set difference and set intersection operators, could produce incorrect results.
- BUG ID 50070 - PackageSuperToolAPI: Right-clicking the grand-child of an adopted package raised an exception.
- BUG ID 50328 - Callback functions of type **postLookFileBake** received an incorrect **abortCallback** argument.
- BUG ID 50525 - The Sphinx-powered HTML documentation made external HTTP requests.
- BUG ID 50628 - Batch rendering an ImageWrite node produced no output files.
- BUG ID 50635 - When choosing the **Copy from Monitor ROI** or **Copy to Monitor ROI** commands in the menu of the **cropWindow** parameter of a RenderSettings node, an exception was raised.
- BUG ID 50680 - Closing the **Curve Editor** or **UV Viewer** tab disabled the display of RGB data in the **Monitor** tab until the end of the Katana UI session.
- BUG ID 50695 - Clockwise mesh vertex winding order was not respected by the Viewer during face set selection, nor by the **Place Specular** Viewer Manipulator.
- BUG ID 50708 - The **isDynamicArray** hint was ignored if provided in an **.args** file.
- BUG ID 50727 - LookFileBake could generate incorrect Look Files when two locations being saved to a Look File had the same set of attributes stored under different names.
- BUG ID 50739 - When attempting to launch Katana while the TMPDIR environment variable was set to the path of a nested, non-existing directory, an exception was raised and Katana failed to launch.
- BUG ID 50819 - Generating a renderer debug output file and then deleting the generated output from disk prevented generating further debug output files in that Katana session.
- BUG ID 50823 - The **Alt**+**M** keyboard shortcut for toggling thumbnails of selected nodes in the **Node Graph** tab did not work for Render nodes.

  As part of a fix for this issue, the **Toggle Thumbnail State of Selected Nodes** command in the **Edit** menu of the **Node Graph** tab has been renamed to **Toggle Thumbnails of Selected Nodes**.

- BUG ID 50921 - When locking a Group node, only the immediate child nodes of the Group node were locked by their parent, and nodes in nested Group nodes could still be edited.
- BUG ID 51024 - There were several issues with muting and soloing of items in the classic Gaffer table in the **Parameters** tab:
  - After soloing an item, icons in the mute column were no longer updated.
  - After muting a rig, the mute icons of items underneath were no longer updated correctly.
  - When muting an item underneath a muted rig, the icon for the muted item was not updated correctly.

Known Issues

- BUG ID 50911 - When changing an array parameter's tuple count/size, any corresponding attributes are not properly updated in the **Attributes** tab.
- BUG ID 49625 - Shading node output ports with names other than **out** are duplicated on flushing caches.
- BUG ID 49069 - The **proxies.firstFrame** and **proxies.lastFrame** attributes, used to specify a time range within which a proxy should be shown, do not work correctly.
- BUG ID 49051- The Viewer may lose sync with the **Scene Graph** tab when changes to expansion state are interrupted.
- BUG ID 47853 - Indication of attribute source nodes, such as the yellow glow in the **Node Graph** tab, is not yet implemented in Katana 2.x.
- BUG ID 44318 - Arnold Live Rendering: Lights created, enabled, or renamed during a Live Render session may not affect that session.
- BUG ID 43072- PRMan Live Rendering: The **adjustScreenWindow** parameter in the RenderSettings node is not taken into account for cameras other than the one specified in the RenderSettings node.
- BUG ID 41152 - When editing parameters of a node that is part of a LiveGroup node and reloading the parent LiveGroup node, the UI state of the **Parameters** tab is reset. This includes scroll bar positions, selections of items, and selections of nested tabs (for example **Object**, **Material**, and **Linking** tabs for a Gaffer node).
- BUG ID 41092 - When reloading a LiveGroup node's parameter interface and contents from its source, parameters of child nodes that are edited in floating panes disappear from those panes.
- BUG ID 40709 - The Alembic library does not support multiple processes or thread access to an Alembic file. This means that a crash occurs when modifying an Alembic file outside Katana, while it's loaded in an open Katana scene. To avoid this, you must **Flush Caches** before attempting to update any modified Alembic files.
- BUG ID 40600 - Undoing a revert of an unpublished LiveGroup node does not restore the LiveGroup's editable and modified state.
- BUG ID 40599 - The user parameters of non-editable LiveGroup nodes can incorrectly be edited.
- BUG ID 40598 - Reverting a LiveGroup node does not revert its user parameters.
- BUG ID 40237 - The nodes can be dragged into the Group bubble of a non-editable LiveGroup node.
- BUG ID 39261 - The operations that lock and unlock nodes do not currently create entries in the Undo History, which can lead to an incorrect node graph state when undoing and redoing operations.

- BUG ID 36926 - The **rendererSettings** > **displayOptions** parameter of a RenderOutputDefine node for the PRMan renderer, shown when its **type** parameter is set to **raw**, cannot be set using the **Parameters** tab.

  For a workaround, the UpdatePrmanSettingNodes shelf script can be used. The script, available in the PRMan 17 shelf, creates an AttributeScript node that converts display options, set using string attributes, to group attributes.
- BUG ID 36691 - The state badges are currently shown for attribute values of dynamic array child parameters, even though only their parent array parameter should appear with a state badge.
- BUG ID 36663 - It is not currently possible to sort notifications in the **Notifications** pop-up window.
- BUG ID 36225 - The **UpdatePrmanSettingsNode** shelf script does not currently deal with expression parameters.The locally set parameter expressions on PRMan nodes need re-adjusting after running the shelf script.
- BUG ID 36176 - The 2D node **Disk Render Upstream Render Outputs** option does not use the batch render method, **batchRender**, for upstream render nodes, instead using **diskRender**.
- BUG ID 36170 - The control keys (notably arrow keys) do not function as expected in shell mode.
- BUG ID 34949 - Using Compiz can lead to text fields not receiving focus events correctly due to an incompatibility between Compiz and QT. Depending on your configuration, disabling Compiz **desktop effects** may resolve the problem.
- BUG ID 34870 - Katana doesn't support render output directory creation for **shadow**, **merge**, and **script** output types.
- BUG ID 33242 - Live Rendering is not designed to work with rapid updates to region of interest (ROI), and may cause unexpected behavior.
- BUG ID 31790 - In the **Parameters** tab, setting an expression for an array or a group parameter currently breaks the parameter's UI.
- BUG ID 28549 - The main menu commands that create nodes, notably **File** > **Import...**, **File** > **Import LiveGroup...**, and **Help** > **I want a pony**, do not create the nodes in a Group or LiveGroup node that has been entered in a **Node Graph** tab, instead they create the nodes in the root level of the node graph document.
- BUG ID 22296 - Live Rendering with AOVs is not supported with PRMan 17 and earlier due to a known issue in these versions of the renderer.

  For a workaround, to allow Live Rendering, disable AOVs in interactive renders by applying an **Interactive Render Filter**, using a RenderSettings node with the **interactiveOutputs** parameter set to just **primary**.

Release Notes for Katana 2.0v4

Release Date

11 August 2015

System Requirements

- Katana 2.0v4 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

Supported Renderers

Katana 2.0v4 provides direct support for PRMan 17.0, Arnold 4.2.4.0 and 4.1.3.3 through the supplied renderer plug-ins. Support for more recent versions of these renderers is provided by Pixar through their RenderMan for Katana (RfK) plug-in, and by Solid Angle through their Arnold for Katana (KtoA) plug-in, respectively. Other renderers are also supported through plug-ins supplied by their respective vendors. To access and receive support for third-party plug-ins, please contact the vendor directly.

The renderer plug-ins provided with Katana are compiled and tested against the above renderer versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API-compliant with the supported versions.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in. To expose new features and options, you may need to modify the renderer plug-in. Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior.

NOTE: We can only guarantee to respond to Katana bug reports when they are reproducible with the provided renderer plug-in, compiled with the supported version of GCC.

Third-Party Dependencies

Katana 2.0v4 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

Feature Enhancements and Important Changes

- BUG ID 21736 - In RenderSettings nodes, the **overscan** parameter is now defined by four values in pixels: left, bottom, right, and top.

NOTE: Existing RenderSettings nodes whose **overscan** parameter is set using an expression must be manually recreated using individual expressions for each value.

- BUG ID 30170 - The **--crop-rect** command-line option, which allows you to render using a cropped display window in batch mode, was silently ignored.

  The non-functional batch mode command-line option **--setDisplayWindowToCropRect** has been removed.
- BUG ID 47258 - The **Scene Graph** tab's **Show Attribute History...** pop-up now provides a **Show Internal Nodes** option, which when checked, displays history entries for nodes within SuperTools directly, rather than delegating to the SuperTool node.
- BUG ID 49884 - A new function was added to the Parameter Expression API:
  `matchGlobalGraphStateVariable(variableName, pattern, matchMethod)`

  All arguments are of type string. The function returns **True** if the value of the Graph State Variable with the given name matches the given pattern using the specified matchMethod, otherwise **False**. Currently matchMethod supports only 'CEL' (CEL matching language).

Bug Fixes

- BUG ID 32793 - The usage information of Katana's command-line interface wrongly referred to a Katana project filename as **compfile**.
- BUG ID 34256 - Preview Rendering 2D nodes failed when **Disk Render Upstream Render Outputs** was set to **Unless Already Cached**. In addition, a crash sometimes occurred when attempting to repeat the Preview Render.
- BUG ID 41052 - The dimensions of Backdrop nodes created through the **Node Graph** tab's **Fit Backdrop Node to Selected Nodes** menu command were not saved to Katana project files.
- BUG ID 48251 - GafferThree's LightPackage has a new method for retrieving the AimConstraint node associated with the light:
  `LightPackage.getAimConstraintNode(forceCreate=True)`
  This method is similar to existing methods on RigPackage:
  `RigPackage.getOrientConstraintNode(forceCreate=True)`
  `RigPackage.getPointConstraintNode(forceCreate=True)`
- BUG ID 48816 - Some minor memory and file descriptors leaks have been fixed.
- BUG ID 48817 - FnAsset: A memory leak in **FnDefaultAssetPlugin** has been fixed.
  For a C++ plug-in to benefit from the fix, it's necessary to recompile it against the latest **FnDefaultAssetPlugin** sources, distributed in **$KATANA_HOME/plugin_apis/**
- BUG ID 48851 - Keyboard focus was lost when editing parameters of a GafferThree node in the **Parameters** tab and using the **Tab** key to change focus from one parameter widget to the next.
- BUG ID 48873 - **GetGlobalAttr()** returned inconsistent results depending on the number of components in the dot-delimited attribute path passed to it.

  When querying for all global attributes through **GetGlobalAttr("")**, some inheriting attributes from ancestor locations were omitted if they existed alongside a non-inheriting attribute.

  When querying for a non-top-level global attribute through **GetGlobalAttr("a.b")**, results were computed using a traversal algorithm different to that in use in the **Attributes** tab, resulting in different results in some cases.

  If you use C++ plug-ins that call **GetGlobalAttr()** you need to recompile your plug-ins to take advantage of these bug fixes.
- BUG ID 48923 - OpScript: Invoking binary operators on Imath types crashed Katana when the left-hand operand was 'nil' or when performing elementwise division and a divide-by-zero occurred.
- BUG ID 49039 - In GafferThree nodes, keyframe values were not correctly set on parameters with the Auto Key option enabled, when their values were interactively set using Viewer manipulators.
- BUG ID 49050 - Scene graph hierarchy expansions started from the **Viewer** tab, by double-clicking on a location's bounding box, were not interrupted when the **Stop Processing** button in the **Scene Graph** tab was pressed.
- BUG ID 49160 - The GafferThree **Package** classes' **getPackageFromNode()** function did not return the enclosing **Package** instance for the given node instance.
- BUG ID 49276 - Locations that had been shown in the GafferThree object table in the **Parameters** tab were still cooked even though the respective GafferThree node parameters had not been edited, resulting in unnecessary Geolib3 processing.

- BUG ID 49277- All immediate children of top-level locations shown in a GafferThree object table were cooked, even if they were not expanded to, resulting in possibly unnecessary Geolib3 processing.
- BUG ID 49387 - CEL expressions that reference a collection located at a non-root location took significantly longer to evaluate than those located at **/root**.
- BUG ID 49449 - Rendering polymesh or subdmesh locations having the **vertexList** attribute specified using a tuple size greater than 1 resulted in an error in the PRMan procedural plug-in.
- BUG ID 49478 - RendererInfo plug-ins: Katana did not fully initialize plug-in instances in all cases before calling methods like **buildRendererObjectInfo()**.
- BUG ID 49649 - Syncing of global selection state to the **Scene Graph** tab (including when the tab is shown after being hidden) took a long time if the selection already contained many locations.
- BUG ID 49683 - The **Select in Scene Graph** action in the **Monitor** tab didn't properly expand the target locations in the **Scene Graph** tab when the scene graph was hidden.
- BUG ID 49982 - Children of **SceneGraphView** widget top-level locations were unnecessarily cooked in some circumstances.
- BUG ID 49986 - Some cases in which Katana accessed invalid memory have been fixed.
- BUG ID 50092 - Katana's command-line help text has been expanded to show the format accepted by the **-t** option.
- BUG ID 50241 - Live Rendering a project using Interactive Render Filters that exist inside a Group node created erroneous connections and could cause subsequent renders to fail.
- BUG ID 50273 - AttributeScript: **GetChildNames()** returned the names of pruned (or otherwise deleted) child locations in addition to existing child locations.
- BUG ID 50338 - The file sequence browser failed to show any sequences if the browsed directory contained more than 10 sequences.
- BUG ID 50368 - Keyboard shortcuts were not detected by widgets under the pointer after closing a menu or pop-up menu. For example, the **Node Graph** tab's Layered Menu keyboard shortcuts did not work after closing the **Find Nodes** pop-up.
- BUG ID 50370 - The CEL form widget didn't respect the **label** hint. It always showed the name of the value policy/parameter.
- BUG ID 50431 - Closing a **Viewer** tab set to the **Four Panes** Viewer Layout caused Katana to crash.

Release Notes for Katana 2.0v3

Release Date

20 July 2015

System Requirements

- Katana 2.0v3 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

Supported Renderers

Katana 2.0v3 supports PRMan 17.0, Arnold 4.2.4.0, and Arnold 4.1.3.3. The supplied renderer plug-ins are compiled and tested against these versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API compliant with the supported versions.

Support for PRMan 18.0 and above is provided directly by Pixar using their RenderMan Studio for Katana plug-in. To get access to RenderMan Studio for Katana please contact Pixar directly.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in.

To expose new features and options, you may need to modify the renderer plug-in.

Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior. Please note that we can only guarantee to respond to Katana bug reports when they are reproducible with the supplied versions of the renderer plug-in, compiled with the supported version of GCC.

NOTE: Currently, it is not possible to load more than one plug-in version of the same renderer. For example, it's not possible to load the plug-in for PRMan 17.0 and PRMan 18 at the same time. However, it is possible to load PRMan 17.0 and Arnold 4.1.3.3.

Third-Party Dependencies

Katana 2.0v3 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

Feature Enhancements and Important Changes

There are no feature enhancements or important changes in this release.

Bug Fixes

- BUG ID 50139 - CEL erroneously reported a successful match when evaluating certain types of union expressions against a location path.

Release Notes for Katana 2.0v2

Release Date

2 July 2015

System Requirements

- Katana 2.0v2 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

Supported Renderers

Katana 2.0v2 supports PRMan 17.0, Arnold 4.2.4.0, and Arnold 4.1.3.3. The supplied renderer plug-ins are compiled and tested against these versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API compliant with the supported versions.

Support for PRMan 18.0 and above is provided directly by Pixar using their RenderMan Studio for Katana plug-in. To get access to RenderMan Studio for Katana please contact Pixar directly.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in.

To expose new features and options, you may need to modify the renderer plug-in.

Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior. Please note that we can only guarantee to respond to Katana bug reports when they are reproducible with the supplied versions of the renderer plug-in, compiled with the supported version of GCC.

NOTE: Currently, it is not possible to load more than one plug-in version of the same renderer. For example, it's not possible to load the plug-in for PRMan 17.0 and PRMan 18 at the same time. However, it is possible to load PRMan 17.0 and Arnold 4.1.3.3.

Third-Party Dependencies

Katana 2.0v2 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

Feature Enhancements and Important Changes

- Geolib3 runtime performance has been improved for clients observing many locations during continuous updates.
- OpScript: The C++ function **GetTransformedBoundAttr()** has been exposed to OpScript as **Interface.GetTransformedBoundAttrPoints()** and **Interface.GetTransformedBoundAttrMinMax()**. These correspond to the two C++ variants of the function that differ in how they represent the transformed bounding box. Further information is available in the OpScript documentation under **Help** > **Documentation** > **OpScript**.
- Documentation of the CookInterface plug-in API headers has been improved.
- BUG ID 30170 - The **--crop-rect** command-line option, which allows you to render using a cropped display window in batch mode, was silently ignored.
- The non-functional batch mode command-line option **--setDisplayWindowToCropRect** has been removed.
- BUG ID 35974 - The **Crash File Selector** dialog has been revised and includes the following improvements:
    - The dialog is now parented to the splash screen, it uses the Katana application icon, and is initialized to the same size as the splash screen.
    - The columns in the dialog's tree widget now use sensible sizes, so that the age of crash files is visible in the dialog without having to scroll horizontally.

  - A **Quit** button has been added to the dialog, which when clicked terminates the application. The application is also terminated when you click the close button in the dialog's title bar, when you press **Alt**+**F4**, or when you choose the **Close** command from the dialog's window menu.
  - When restoring a Katana project from a crash file, appropriate info messages are logged.
- BUG ID 36610 - TimeOffset nodes did not affect animated Viewer proxies.
- BUG ID 36979 - Generating render debug output now prints progress and error messages to the console.
- BUG ID 43354 - When changing parameters of nodes while the **3D Update Mode** was set to **Manual**, any changes made in the generated scene graph were not reflected in subsequent renders.

  3D node parameter values are now finalized with all pending changes prior to performing a render, as part of calls of **Nodes3DAPI.GetRenderOp()**.

  Any custom UI element that delays submission of 3D node parameter changes may register a callback of type **finalize3DNodeChanges** as follows:

```
from Katana import Callbacks

def finalize3DNodeChanges(**kwargs):
    # Submit all pending changes

Callbacks.addCallback(Callbacks.Type.finalize3DNodeChanges,
                      finalize3DNodeChanges)
```

- BUG ID 48043 - Calling **reset()** on a Python asset plug-in now executes on all the instances of the plug-in, even the ones living in other processes. Furthermore, the function **AssetAPI.ResetAllAssetAPIPlugins()** now correctly invokes **reset()** on all the Python asset plug-ins.
- BUG ID 48173 - OpScript: Objects provided by the **FnAttribute** module, including instances of types such as **IntAttribute** and **FloatAttribute**, now respond to **tostring()** by returning an abridged representation of the object that is less verbose than **getXML()**. The representation is vaguely Lua-inspired, but is intended for debugging, not serialization, for example:

```
print(myDoubleAttr)
-- prints the following:
<DoubleAttribute: values=11, samples=1, tupleSize=1> {
  [0.0] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0,  ... },
}
```

- BUG ID 48456 - It is now possible to retrieve the Group node that has been entered in a **Node Graph** tab by calling **getEnteredGroupNode()**, for example:

```
UI4.App.Tabs.FindTopTab('Node Graph').getEnteredGroupNode()
```

- BUG ID 48885 - The following new types of callbacks have been added:
  - **onMasterMaterialAssigned** - executed when assigning or unassigning a Master Material to, or from, a light in GafferThree. Receives two arguments: a **LightPackage** instance that represents the light to which a Master

Material has been assigned, and a **MasterMaterialPackage** instance that represents the Master Material that has been assigned to the light. If a Master Material is unassigned, that second argument is **None**.

- **onLookFileMaterialSet** - executed when setting a Look File Material on a light or Master Material in GafferThree. Receives three arguments: a **Package** instance that represents the light or Master Material for which a Look File Material has been set, the filename or asset ID of the Look File that is referenced, and the path of the material that is referenced within that Look File.
- **onLookFileMaterialActiveSet** - executed when activating or deactivating a Look File Material on a light or Master Material in GafferThree. Receives two arguments: a **Package** instance that represents the light or Master Material for which a previously set Look File Material has been activated or deactivated, and a boolean flag indicating whether it has been activated or deactivated.

- BUG ID 48925 - Support for displaying the cone of directional quad lights has been added to the default light viewer modifier.

Bug Fixes

- Geolib3 runtime clients could receive 'location exists' events for locations that had since been deleted due to a change committed during scene cooking.
- BUG ID 21735 - With certain graphics hardware and drivers, when selecting the **Shaded (filmlook)** option in the **Viewer** tab's **Display** menu, an OpenGL-related error occurred and nothing was displayed in the viewport.
- BUG ID 34285 - Dragging the node icon of a **Node Drop Proxy** parameter onto the **Drop Node Here** label of the same parameter caused Katana to crash.
- BUG ID 37080 - Existing Live Render sessions were not canceled correctly when opening a new project.
- BUG ID 39850 - Setting the **geometry.voxelResolution** attribute of a scene graph location of type **volume** to **(0, 0, 0)**, and then expanding the scene graph caused Katana to crash.
- BUG ID 44195 - Deferred Ops, such as AttributeScript and OpScript nodes set to non-immediate modes, could not modify the **ops** attribute group during resolution, and therefore could not register additional deferred Ops.
- BUG ID 46095 - The code completion pop-up in the **Python** tab did not use case-insensitive sorting for the suggestions.
- BUG ID 46097 - The code completion suggestion of the **Python** tab, in IDE mode, would sometimes not respond to scrolling through the completion options.
- BUG ID 46625 - Using the **Parameters** tab to set a Group parameter to a Python expression that referenced the same parameter caused Katana to crash.
- BUG ID 46901- Graphviz libraries were loaded by renderboot at render time.
- BUG ID 47207 - Using Unicode character keyboard shortcuts, such as **AltGr**+**A**, with the pointer over the **Node Graph** tab raised a UnicodeEncodeError exception.
- BUG ID 47833 - Querying attribute history in a node graph containing a Fork3D node failed to identify contributing nodes upstream of the Fork3D node and resulted in an incorrect nodes list.

- BUG ID 48049, 48050 - When making changes to a LiveGroup node's parameter or contents, not all operations resulted in the LiveGroup node being marked as modified, potentially resulting in data loss when closing a Katana project without being prompted to save LiveGroup changes.
  Operations that did not mark a LiveGroup node as modified included: creating, renaming, and removing input or output ports, connecting or disconnecting nodes inside, setting a child node's color or comment, and modifying a Backdrop node's text.
- BUG ID 48061 - When the **Dim Nodes Not Contributing to Viewed Node** option was turned on, changing the current frame in the Katana timeline did not update the dimming of nodes upstream of the currently viewed nodes.
- BUG ID 48158 - When restoring a Katana project from an autosave file after a crash, all LiveGroup nodes in the project were made editable, regardless of their original states.
  Now, LiveGroup nodes are restored to the original states they were in at the time when the autosave file was created.
- BUG ID 48248 - Importing rigs into GafferThree nodes containing adopted packages failed to warn the user about naming collisions.
- BUG ID 48263 - When turning **useLookFileMaterial** off for a light location with a Look File Material assigned on the **Material** tab in the parameter interface of GafferThree nodes, if the light's shader had a parameter of type **color** and if the color parameter group was open, an exception was raised.
- BUG ID 48576 - The **editLookfileMaterial** parameter on the **Material** tab in the parameter interface of GafferThree nodes for adopted packages has been renamed to **editLookFileMaterial**, with a capital **F**.
- BUG ID 48584 - Moving the pointer over code in the **Python** tab immediately displayed code inspection tooltips, rather than after a short delay.
- BUG ID 48617 - The error message generated by custom nodes that are loaded in the absence of their defining plug-in has been modified for clarity, referencing both node name and type.
- BUG ID 48619 - Details on how to instantiate Attribute objects from within OpScript have been added to the OpScript Reference.
- BUG ID 48776 - The **ViewerModifierInput::getUniversalAttribute()** function has been deprecated.
- BUG ID 48804 - CEL expressions containing more than one recursive pattern (**//**) could generate incorrect results.
- BUG ID 48812 - When assigning a Master Material to a light that used a Look File Material, the Look File Material was not deactivated.

  When assigning a Look File Material to a light that used a Master Material, the Master Material was not unassigned.
- BUG ID 48854 - The OpScript documentation referred to some constants that were only available in the C++ API. These constants, **kFnKatGeolibDefaultInput** and **kFnKatGeolibNullInput**, are now available in OpScript as **Interface.InputIndex.Default** and **Interface.InputIndex.Null**.
- BUG ID 48949 - When performing a render, the value of the **KATANA_RESOLUTIONS** environment variable was changed and not properly reset.
- BUG ID 48977 - Clicking the **Force Refresh** button in the parameter interface of a shading node that was in an error state, occasionally raised an exception.

- BUG ID 48999 - The global Graph State Variable pop-up widget, activated from the main menu bar, did not provide a scroll bar when the number of variables exceeded the maximum number of elements that could fit in the pop-up widget's height.
- BUG ID 49021 - **Collect and Select...** and **Collect and Display Results...** actions for non-root collections in the **Scene Graph** tab only worked for collections sourced from a Look File.
- BUG ID 49022 - Collection attributes set by CollectionCreate nodes were wrongly inherited by descendant locations.
- BUG ID 49055 - Calling **checkDynamicParameters()** on a node could, if the node was in error state, raise an exception.
- BUG ID 49092 - AttributeCopy would delete any attribute whose source attribute did not exist.
- BUG ID 49102 - Viewer objects corresponding to scene graph locations that are deleted while the Viewer was hidden could remain visible once the Viewer was shown, and the drawing of their GL representations could cause Katana to crash.
- BUG ID 49146 - Deferred AttributeScript user parameters baked into Look Files from Katana 1.x were missing when brought into Katana 2.0.
- BUG ID 49147 - The (undocumented) **InterfaceUtils.BuildLocalsAndGlobals()** OpScript function would return **nil** instead of returning a pair of the local and global attributes for the current location.
- BUG ID 49216 - The Viewer proxy **proxies.currentFrame** attribute wasn't handled properly.
- BUG ID 49247 - The options listed for the **application/fontFamily** preference in Katana's **Preferences** dialog included names of font families of non-Latin writing systems, including names of symbol fonts.
- BUG ID 49253 - When creating **QComboBox** widgets and making them editable in custom dialogs from within Katana, their integrated **QLineEdit** widget did not use the application's font as specified in Katana's preferences.
- BUG ID 49255 - The shader selection pop-up widget that can be opened from the GafferThree object table could extend off the edge of the screen, depending on the position of the **Parameters** tab.
- BUG ID 49276 - Locations that had been shown in the GafferThree object table in the **Parameters** tab were still being cooked, even when parameters of the respective GafferThree node were no longer being edited, resulting in unnecessary Geolib3 processing.
- BUG ID 49277 - All immediate children of top-level locations shown in a GafferThree object table were cooked, even if they were not expanded to, resulting in possibly unnecessary Geolib3 processing.
- BUG ID 49306, 49307 - Loading a scene graph bookmark would reset **Open**, **Selection**, and **Pinned** states if they were not saved in the bookmark.
- BUG ID 49315 - The description of the **node_setParent** event type in the developer documentation of the **NodegraphAPI** Python package was inaccurate, and descriptions for the **node_renameInputPort**, **node_renameOutputPort**, **port_connect**, and **port_disconnect** event types were missing.
- BUG ID 49343 - When saving a Katana project containing editable LiveGroup nodes that used the same source as non-editable LiveGroup nodes in the same project, the parameters and/or contents of those editable LiveGroup nodes were not saved to the **.katana** file, potentially leading to loss of data in case of a crash.
- BUG ID 49355 - The **viewer > interactiveProcessingDelay** preference was ignored on startup if set to zero.
- BUG ID 49369 - The **viewer > interactiveProcessingDelay** preference was used only for 'look-through' manipulation. It now applies to all Viewer Manipulators.

- BUG ID 49373 - When calling **NodegraphAPI.LiveGroup.LockAllLiveGroups()** on a Group node, the contents of editable LiveGroup nodes that had a source set for them inside of the given Group node were wrongly locked.
- BUG ID 49392 - When publishing and finishing the editing of contents of LiveGroup nodes, child nodes inside the group that had been added through scripting were not properly locked.
- BUG ID 49394 - When calling **NodegraphAPI.LiveGroup.LockAllLiveGroups()** on a Group node, the contents of LiveGroup nodes nested within editable LiveGroup node inside the given Group node were not locked properly.
- BUG ID 49432 - The introduction of support for editing LiveGroup nodes made it possible to edit a nested LiveGroup without making its ancestor LiveGroup nodes editable first, however, the following two specific cases did not work correctly:
    - When attempting to lock a modified LiveGroup node using the Python API, a call of the **LiveGroupNode.setLocked()** function was wrongly rejected with a warning saying that Katana was **Unable to lock LiveGroup node "<name>": The node is in an editable state, and its contents have been modified**. It is now possible to lock modified LiveGroup nodes.
    - When locking an editable, but unmodified LiveGroup node using the Python API, a call of the **LiveGroupNode.setLocked()** function resulted in the node being made non-editable. Now, when calling **setLocked()**, the editable state of LiveGroup nodes is not affected in any case.
- BUG ID 49541 - When using a Viewer Manipulator for a light that is adopted in a GafferThree node after being created upstream of that node, interactive edits of viewer manipulator handles could be recorded in a different GafferThree node.
  This was due to the **attributeEditor.material.exclusiveTo** attribute on the corresponding light location pointing to the original Material node, and not the Material node in the GafferThree node in which the light is adopted.
- BUG ID 49610 - Attempting to acquire the Python GIL in a plug-in's **flushCaches()** function could result in deadlock.

Release Notes for Katana 2.0v1

Release Date

14 May 2015

System Requirements

- Katana 2.0v1 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

Supported Renderers

Katana 2.0v1 supports PRMan 17.0, Arnold 4.2.4.0, and Arnold 4.1.3.3. The supplied renderer plug-ins are compiled and tested against these versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API compliant with the supported versions.

Support for PRMan 18.0 and above is provided directly by Pixar using their RenderMan Studio for Katana plug-in. To get access to RenderMan Studio for Katana please contact Pixar directly.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in.

To expose new features and options, you may need to modify the renderer plug-in.

Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior. Please note that we can only guarantee to respond to Katana bug reports when they are reproducible with the supplied versions of the renderer plug-in, compiled with the supported version of GCC.

NOTE: Currently, it is not possible to load more than one plug-in version of the same renderer. For example, it's not possible to load the plug-in for PRMan 17.0 and PRMan 18.0 at the same time. However, it is possible to load PRMan 17.0 and Arnold 4.1.3.3.

Third-Party Dependencies

Katana 2.0v1 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

Feature Enhancements and Important Changes

GafferThree

A new Gaffer node type, named GafferThree, has been created to provide improved performance when dealing with large numbers of lights in Katana projects. The GafferThree implementation takes full advantage of the new scene graph processing library Geolib3. The existing legacy Gaffer node type from Katana 1.x is still present, and previously created projects should continue to work, but it is advisable to move to using the new GafferThree node type where possible.

NOTE: For more information, see the *What's New in Katana 2.0* document that accompanies this release.

NOTE: All items in the GafferThree object table now require a double-click to edit, including the shader cell. Note that this differs from the Gaffer shader cell, which is still editable with a single-click.

Graph State Variables

Node types and UI elements have been added to support the setting and querying of new, node graph-level Graph State Variables. These can be used to control which nodes in the node graph contribute to scene graph processing at a particular time. This can greatly simplify multi-pass/layer workflows or any other tasks that might want to re-use node setups with different inputs or outputs.


NOTE: For more information, see the *Graph State Variables* chapter of the *Katana User Guide*.

Light Linking

- BUG ID 28741 - Baking lights was breaking light linking as the paths to the lights are stored as absolute. Now the LookFileBake node automatically stores the light linking information needed and the LookFileLightAndConstraintActivator node corrects the absolute paths.

  The LightLinkSetup node has been modified to support appending light linking information to the incoming scene by default. You can choose to override the incoming settings if needed by appropriately modifying the new **action** parameter on the node.
- BUG ID 46284, 47647, and 47648 - Enhanced support for setting the initial value for a light's illumination or shadow visibility in its entry in the **light list**: a new **initialState** parameter is present in the LightLinkSetup node, as well as the **linking** tab for lights in GafferThree.

  For clarity, the LightCreate node's **lightListDefaults.enable** parameter has been renamed to **initialState**.

Live Rendering

- The **Live Render Control** tab has been replaced with several more Katana-native, more flexible features. These include:
  - A new column in the **Scene Graph** tab that allows toggling locations to generate Live Rendering updates when their attributes change.
  - GenericAssign-based node types named **<renderer>LiveRenderSettings** that are used to set settings for Live Rendering.
  - Qt menu actions have replaced the custom Live Rendering command buttons that were previously implemented in the renderer info plug-ins and displayed in the **Live Render Control** tab.
  - The option to use the same camera as the **Viewer** tab has been replaced by a button at the bottom of the **Viewer** tab, named **Live Render from Viewer Camera**.


NOTE: For more information on these see the *What's New in Katana 2.0* document.

- A new LiveRenderAPI Python package provides access to several functions that allow users to modify the behavior of the Live Rendering system. These include:

  - **SendCommand()** - Sends custom Live Render commands to the render plug-in via the command socket.
  - **SendData()** - Sends custom data updates to the render plug-in via the data socket.
  - **AppendTerminalOp()** and **RemoveTerminalOp()** - Adds or removes additional terminal Ops to the Live Rendering client allowing you to customise the scene graph data that is passed through to renderers.
  - **ClearAllTerminalOps()** - Removes all Live Rendering terminal Ops, including the default (specified in renderer info plug-ins).
  - **RestoreDefaultTerminalOps()** - Restores the default terminal Ops.
- Rebinding shaders during a PRMan Live Render now happens automatically on update without the need to manually activate from the **Live Render** menu.

Ops

- The AttributeModifierResolve node type has been deprecated in favor of OpResolve. Existing projects are still able to load and use AttributeModifierResolve nodes, but users are no longer able to create them in the UI. OpResolve does the same job as AttributeModifierResolve.
- Ops can be used to define viewer proxies on scene graph locations. Two main attribute conventions are currently supported: ViewerProxyLoader (legacy mode) and Op-based. For more information on them and examples see the *Viewer Proxies* section in the *What's New in Katana 2.0* document.
- BUG ID 39971 - The options for the **applyWhen** parameter are now more consistent across the following node types:
  - AttributeFile_In
  - AttributeModifierDefine
  - AttributeScript
  - GenericOp
  - OpScript

  The following **applyWhen** options have been renamed:
  - AttributeModifierDefine - **deferred** is now **during op resolve**
  - AttributeScript - **during attribute modifier resolve** is now **during op resolve**
  - AttributeFile_In - **deferred** is now **during op resolve** and **during Katana standard resolve** is now **during Katana look file resolve**.

Python Tab

- Preferences for the **Python** tab in the **python** category of preferences have been revised:
  - The **commandFont** and **resultFont** preferences have been removed, as it was not possible to change them from the **Preferences** dialog. The appearance of text in the Python tab now depends on the application font preference.
  - Preferences have been added to control the auto-completion behavior (**autoCompletionBehavior**) as well as whether or not to show help tooltips in the **Python** tab (**showHelpTooltips**).

- The **Python** tab has been enhanced to support better syntax highlighting, code completion, introspection info using tooltips, and block indentation and un-indentation.

API Changes

- Katana has been upgraded to use Python 2.7.3 and Qt 4.8.5, matching the VFX Platform CY2014 specification.
- CEL pattern extensions provide a way of matching children of the locations matched by a CEL expression. Commonly this is used to take a collection that already exists (for instance, root locations of certain assets) and create a CEL statement that matches specified child locations of that collection.

NOTE: The undocumented CEL **APPLY()** operator has been removed in favor of CEL Pattern Extensions, see the *Katana User Guide* for more information.

- The signature of the `createProxyAttr()` method of `BaseProxyLoader` has been modified to return a GroupAttribute that sets up the execution of an Op instead of a Scene Graph Generator. This Op is specified by the child **opType** StringAttribute and optional child **opArgs** GroupAttribute. The old signature is still supported for legacy implementations but may be deprecated in the future.
- Attribute History: There is now an API for querying Attribute History from Python. This can be found in the `UI4.Util.AttributeHistory` module. Attribute History can be queried synchronously, in which case the UI will block until the result is computed and returned, or asynchronously if users provide a callback to run when the computation is complete.
- BUG ID 35076 - Katana previously shipped with a build of **OpenColorIO** that used **FnOpenColorIO** namespaced symbols, but the **OpenColorIO** libraries were not named accordingly (**libOpenColorIO.so** and **PyOpenColorIO.so** Python bindings). This caused problems with using a custom facility-installed **OpenColorIO** in parallel with the Katana libraries. This has been updated so that the libraries shipped with Katana are now Fn-prefixed too (**libFnOpenColorIO.so** and **FnPyOpenColorIO.so**). Using the Python bindings is still possible through **import FnPyOpenColorIO**
  -Any code using **PyOpenColorIO** will need updating to either use **FnPyOpenColorIO**, or a facility-installed **OpenColorIO** could be used instead.
- BUG ID 38948 - The screen window of an orthographic camera is now computed within the CameraSettings API. This calculation has been removed from the PRMan plug-in.
- BUG ID 41504 - The **GetAttr()** AttributeScript function now takes an optional argument, **asFnAttribute**, which, if **asAttr** is set to **True**, determines whether the attribute is returned as a **PyFnAttribute** or a **PyScenegraphAttr**. This argument defaults to **False** for backwards compatibility.
- BUG ID 43300 - The **imageInfo** parameter passed to **FnRenderOutputLocationPlugin::computeFilePath()** now includes the following new fields:
  - **scenePassName** - a string including the scene file name and the Render node's **passName** parameter's value, for example, mySceneFileName_BeautyPass. Notice that this is the same value set for **passName**.
  - **renderPassName** - the Render node's **passName** parameter's value, for example, BeautyPass.
  - **sceneName** - the scene file name, for example, mySceneFileName.

- BUG ID 43302 - RenderOutputLocation plug-in APIs have been extended to allow the **FnRenderOutputLocationPlugin::computeFilePath()** and **FnRenderOutputLocationPlugin::computeLocation()** methods to access the renderSettings group attribute stored on the /root location of the scene graph.

  The methods' signatures have changed as shown below:

```
virtual std::string computeFilePath(
    const FnAttribute::GroupAttribute & outputAttrs,
    const FnAttribute::GroupAttribute & locationAttrs,
    const FnAttribute::GroupAttribute & renderSettingsAttrs,
    const FnAttribute::GroupAttribute & imageInfo,
    bool makeVersionTemplate) = 0;
virtual std::string computeLocation(
    const FnAttribute::GroupAttribute & outputAttrs,
    const FnAttribute::GroupAttribute & locationAttrs,
    const FnAttribute::GroupAttribute & renderSettingsAttrs,
    const FnAttribute::GroupAttribute & imageInfo) = 0;
```

NOTE: Custom render output location plug-ins have to be recompiled in order to be compatible with the new signatures.

- BUG ID 43352 - The **Render** plug-in API has been renamed to **FnRender** for consistency with other Katana plug-in APIs.
- BUG ID 46829 - The following methods to resolve and query information about asset IDs are now available in AttributeScript nodes through the **DefaultAssetPlugin** module:

```
DefaultAssetPlugin.isAssetId(string)
DefaultAssetPlugin.containsAssetId(string)
DefaultAssetPlugin.resolveAsset(assetId)
DefaultAssetPlugin.resolvePath(path, frame)
DefaultAssetPlugin.getUniqueScenegraphLocationFromAssetId(assetId, includeVersion)
DefaultAssetPlugin.getRelatedAssetId(assetId, relation)
```

Miscellaneous

- Proxy caches are now considered animated by default. Static proxy caches can be defined by setting the **proxies.static** IntAttribute to 1.
- The internal Katana documentation has been replaced with Sphinx-powered, searchable docs that are now opened in an external web browser. As part of this work, the **Help** tab has been deprecated, and the **Example Projects** tab has been added.
- A number of node types that were undocumented and unavailable in Katana's UI have been deprecated in Katana 2.0v1: AsmbCmpt_ PostProcess, BlockerCreate, F3D_In, GenericGeo, HDF_In, IncomingTest, NetworkAttributeScript, NetworkAttributeScriptInput, NetworkAttributeScriptNode, NetworkMaterialEdit,

ProceduralSetup, Resolve, ScenegraphLocationModifierDefine, ScenegraphLocationModifierResolve, ShadowRenderAssign, TextureResolve, and TxasnMappingInfoApply.

- The VelocityApply node type has been modified to add the following two parameters:
  - **velocityAttribute** - The name of the attribute representing the velocity information to be used by the node. If the parameter is not set, the following attributes are checked:
    - **geometry.point.V**
    - **geometry.point.v**
    - **geometry.arbitrary.v**
  - **velocityUnits** - Units to be used to interpret the values stored in the velocity attribute, with the following options:
    - **units / second**
    - **units / frame**
- The way that Sky Dome items are implemented in classic Gaffer has changed. Instead of an **arnoldSurfaceShader** of type **skydome_light** on the item's Material node, materials on Sky Domes are now resolved internally in Gaffer.
- The RendererProceduralArgs node in create mode now has namespace support, similar to the namespace support already in the Material node.
- BUG ID 37514 - The **adjustScreenWindowWhen** parameter has been added to RenderSettings nodes to control when the **adjustScreenWindow** parameter is evaluated. By default, the **adjustScreenWindow** parameter is now evaluated in a deferred manner by a separate **AdjustScreenWindowResolve** Op as an implicit resolver. This allows the **renderSettings.adjustScreenWindow** attribute that is stored on the **/root** location to be modified later in downstream nodes before it takes effect. You can set **adjustScreenWindowWhen** to **immediate** to have the **adjustScreenWindow** parameter take effect immediately.
- BUG ID 38482 - A new keyboard shortcut, **Ctrl+F**, has been added to the **Node Graph**, **Parameters**, **Scene Graph**, **Attributes**, and **Project Settings** tabs to open the **Find** pop-up.
- BUG ID 40371 - An **adjustAncestors** parameter has been added to the BoundAdjust node to enable/disable the adjusting of ancestor bounds.
- BUG ID 41293 - The node type name of Backdrop nodes has been renamed from **BackdropNote** to **Backdrop**. This is to achieve consistency with other products at The Foundry, such as NUKE.
- BUG ID 41366 - The Alembic_In node now supports a **useOnlyShutterOpenCloseTimes** parameter that forces the Alembic cache to only use the time samples corresponding to shutter open and close times when the **maxTimeSamples** parameter is set to **2**. The parameter is available in the **advanced** section in the **Parameters** tab. Note that the useOnlyShutterOpenCloseTimes argument is also supported by the AlembicIn Op.
- BUG ID 41586 - The **Ctrl**+middle-mouse button drag and drop from the **Attributes** tab into the **Python** tab now produces an explicit PyFnAttribute representation.Using the **Ctrl**+**Shift**+middle-mouse button generates the old PyScenegraphAttr code.
- BUG ID 42168 - The attribute convention for referenced materials has changed from **reference.lookfile** to **reference.asset**. MaterialResolve (both the Op and DAP) checks both conventions in order to handle, for example, a baked scene containing **reference.lookfile**. The nodes that create a referenced material location (LookFileMaterialIn, for example) follow the new convention.

- BUG ID 42439 - The pixel probe toolbar in the **Monitor** tab has been modified to show a luminance component for sampled pixels next to the existing RGBA components. Also, the menu buttons for selecting a scene graph location in the **Scene Graph** tab and for copying the path of a scene graph location are now only shown when a scene graph location corresponds to the object under the pointer, as determined through an ID pass, which can be turned on using the **Render** > **3D Rendering** > **Render ID Pass** toggled menu item.
- BUG ID 42381- The **application** > **rendering** > **restartLiveRenderOnTimeChange** preference has been added to control whether or not Live Rendering sessions are restarted when changing the current frame.
- BUG ID 44555 - The channel name provided to the renderer-specific OutputChannelDefine nodes may now contain dot (**.**) characters. This is supported through the addition of an exact **name** child parameter of the corresponding, safely-named group attribute under **[renderer]GlobalSettings.outputChannels**.
- Katana now officially supports Arnold 4.2. The included Arnold 4.2 plug-in has been validated against Arnold 4.2.4.0 and has received a number of fixes for compatibility and stability.
- Two example projects (accessible through **Help** > **Example Projects**), demonstrating the capability of Graph State Variables, ship with Katana. These projects show usage of the VariableSet, VariableSwitch, and VariableEnabledGroup nodes, local Graph State Variables, retrieval of variable values from OpScript, and advanced CEL-like patterns for matching variable values.
- Katana ships with a Crowd System example project that highlights many of Katana 2.0's most powerful features, including multi-shot, multi-output workflows enabled through Graph State Variables, and flexible geometry processing in OpScript.
- Katana's existing example projects have been updated to consistently use the new GafferThree and OpScript nodes instead of Gaffer and AttributeScript.
- BUG ID 25170 - Katana ordered auto-saved project files with the oldest displaying first in the **--crash** dialog. The most recently auto-saved project file is now displayed first.
- The Monitor pixel probe colorspace now defaults to OCIO scene linear role, matching the behavior of the image display colorspace.
- The undocumented 'Bake to Material' Katana 1.x feature has been removed.
- BUG ID 32589 - Renders performed within a second of an edit by Viewer manipulation would not include the edit. This time may now be set with the **Viewer** > **interactiveProcessingDelay** preference, and defaults to 0.25 seconds.
- BUG ID 43945 - A new preference has been added, **parameters**>**selectParameterValueOnFirstClick**, that specifies if the value of a text or numeric widget in Katana should be completely selected on mouse click.
- BUG ID 44724 - The appearance of toolbar buttons across Katana's UI has been refined, so that they are easier to click when using larger screen resolutions.

  Notable changes include:

  - the sizes of toolbar buttons on tabs, such as the **Scene Graph** tab, which are now more consistent with one another.
  - moving the pointer over a button, which now highlights the button's clickable area with a background color.
  - the enlargement of several very small buttons on the UI.

Furthermore, tooltips have been added to more toolbar buttons. For example, tooltips can now be found on the **Find Nodes** button in the **Node Graph** tab, the **Find Attributes** button in the **Attributes** tab, and the **Edit Node** and **Edit Parameter Interface** buttons in the **Parameters** tab.

- BUG ID 44725 - Pop-up widgets of toolbar buttons, such as **Shelf Actions**, **Interactive Render Filters**, and **Find** buttons, are now sized according to their initial contents. They can be resized by dragging a size grip in the lower-right corner of the pop-up. Double-clicking the size grip resets a pop-up to its initial size, taking its contents into account.
- BUG ID 44726 - The **nodegraph**>**findOnlyNodesInThisGroupDefault** preference has been added to allow users to control the default state of the **Show Only Nodes in this Group** checkbox in the **Node Graph** tab's **Find** pop-up. The state of the checkbox can still be changed during a Katana session, but is initialized according to the new preference in new sessions of Katana. Please note that due to a current known issue (Bug 46542), this preference may not appear during the first session of this Katana version.
- BUG ID 44922 - Support has been added for V-Ray light representations in light Viewer Modifiers.
- BUG ID 39997 - It is now possible to edit a nested LiveGroup without making its parent LiveGroup nodes editable first.

  The **Edit Parents and Contents** command has been added to the context menu of LiveGroup nodes in the **Node Graph** tab. The command makes all parent LiveGroup nodes of a right-clicked LiveGroup node editable before making the LiveGroup node itself editable.

  When choosing **Publish and Finish Editing Contents...** in the context menu of a LiveGroup node in the **Node Graph** tab, the contents of the node are no longer loaded from the published source right after publishing.
- BUG ID 44916 - LookFileResolve: A new environment variable has been added to enable a fall-back on the default pass when requesting a missing pass:

  `KATANA_LOOKFILE_DEFAULT_PASS_FALLBACK=1`

  When enabled, no error is produced, and the default pass is used. Note that the **lookfile.resolvedPass** attribute always reports the **requested** pass, and is therefore not affected by this fall-back.
- BUG ID 46071 - In **.args** files, tuples of floats and strings are now supported as valid types for values in the **presets** parameter for the **presetsGroups** widget type.
- BUG ID 47318 - The **Render** > **Render View Node** command has been renamed to **Preview Render View Node**, and its default keyboard shortcut has been changed from **P** to **Ctrl**+**P**. You can still use the **P** keyboard shortcut but we encourage you to adopt the new **Ctrl**+**P** keyboard shortcut for starting a Preview Render.

  A command for starting a Live Render of the currently viewed node has been added to Katana's **Render** main menu: **Live Render View Node**. The keyboard shortcut to trigger the command is **Ctrl**+**Shift**+**P**.

  The default keyboard shortcut of the **Render** > **Repeat Previous Render** command has been changed from **** to **Ctrl**+****. You can still use the **** keyboard shortcut but we encourage you to adopt the new **Ctrl**+**** keyboard shortcut for restarting a previously started render.

  The new modifier keyboard shortcuts allow you to start a Preview or Live Render, even when editing text within the application, for example, when writing code in a **Python** tab. (See also: Bug ID 32778 for related information).
- BUG ID 47150 - A dedicated **Reload Args Files** command for flushing the Args file cache and refreshing the UI has been added to the **Util** main menu. The command differs from **Reload Shaders and Args Files** in that the shader cache is not cleared, thus this command is less costly.

- BUG ID 47319 - The **Alt**+**C** keyboard shortcut has been added to the **Render** > **Live Rendering** > **Create Snapshot in Catalog** command.

  The **getActionShortcut()** static method has been added to the **BaseLiveRenderAction** class in the **PluginAPI.BaseLiveRenderAction** Python module. The function can be used in custom Live Render action classes to return the textual form of a keyboard shortcut to use for the respective action in the UI.
- BUG ID 47892 - Look Files containing Master Materials assigned to lights using referential inheritance would not apply correctly to a hierarchy that differed to that of the original. Master Material assignment is now expressed relative to the light, whereas previously the absolute path to the Master Material was specified.
- The Alembic_In Scene Graph Generator has been removed, having been superseded by the AlembicIn Op.
- BUG ID 47444 - The NonpersistantSwitch node type is now available as NonpersistentSwitch. The former spelling of this node type is now deprecated, and may be removed in a future version.
- BUG ID 47687 - The deprecated **FFindPopup** widget class has been removed. **FilterablePopupButton** should be used instead.
- BUG ID 48309 - If the MaterialResolve Op finds an **errorMessage** attribute on the material referenced by a **materialAssign** attribute on a location, it copies the error message over to the **material** attribute on the location to which the material is assigned, creating a **material.errorMessage** attribute on the location.

Bug Fixes

- ImageChannels nodes were missing the multi-input port in the **Node Graph** tab.
- Katana's crash handler did not handle SIGABRT, SIGILL or SIGFPE signals, but rather only SIGSEGV.
- The DependencyMerge node type used legacy names for the **farmSettings** parameter, and the **farmSettings.farmFileName** parameter did not appear correctly for Render nodes.
- BUG ID 27147 - Due to an internal bug in Qt 4.7.2, an error message similar to the following was occasionally displayed in the console: "Application asked to unregister timer 0x13000076 which is not registered in this thread. Fix application." (https://bugreports.qt-project.org/browse/QTBUG-16175).
- BUG ID 28477 - When editing the parameters of a Gaffer node in the **Parameters** tab and using the mouse wheel to scroll with the pointer over the title of a nested tab (**Object**, **Material**, or **Linking**), the tabs under the pointer were switched while also scrolling widgets in the **Parameters** tab in the direction of the wheel. The new behavior is to not switch tabs when using the mouse wheel in the **Parameters** tab, which is consistent with behavior in Nuke.
- BUG ID 32062 - The **Edit Expression...** command in the context menu of parameters in the **Parameters** tab was incorrectly placed between the **Expression** and **Curve** menu items.
- BUG ID 33784 - The LightCreate node's **makeInteractive** parameter was not hinted as boolean.
- BUG ID 35672 - An exception was raised after choosing the **'Find and Select' Selected Items** option from the right click menu for a CEL collection in the **Parameters** tab.
- BUG ID 36422 - The **Messages** pop-up window suffered from poor performance when logging a large number of messages through Python.
- BUG ID 36428 - When pressing certain keyboard shortcuts over some tab types, such as the **Tab** key over the **Node Graph** or **Monitor** tabs while a modal dialog or pop-up widget were shown, the keyboard shortcuts were

incorrectly still dispatched to widgets underneath the pointer, ignoring the modal state of the dialog or pop-up widget.

- BUG ID 36687 - Setting a key frame on a parameter was not indicated by a green background.
- BUG ID 36700 - Local renders failed when the host name of an artist's workstation could not be resolved. Katana now falls back on 'localhost'.
- BUG ID 37477 - Adding an invalid path to the ImageRead node caused Katana to crash.
- BUG ID 38307 - Calling **NodegraphAPI.CreateNode()** from the Python API to create a node created multiple entries in the undo history.
- BUG ID 38752 - Terminating a Katana UI session with a SIGTERM or SIGINT signal during a render didn't properly terminate the render process which continued holding a license.
- BUG ID 38839 - Orthographic camera projection in the Viewer was incorrect.
- BUG ID 40375 - Setting the **crop** parameter of a ZoomToRect node to **Yes** did not have an effect on the rendered image.
- BUG ID 40404 - Katana could become unresponsive for a period of time when enabling a KVM switch.
- BUG ID 40414 - Viewer Modifier Plug-in API: the **setAttributeOverride()** and **setLiveAttributeOverride()** functions have been deprecated.
  Since only one VMP may run on any scene graph location, there is no need to provide overrides outside of the individual plug-in.
- BUG ID 41117 - The **Last File Save** bookmark was not written to disk when a project was first saved.
- BUG ID 41236 - The ShadowManager uses a temporary directory for storing shadow renders, but would only evaluate **KATANA_TMPDIR** when a light was added to a pass, resulting in erroneous render output locations when the project was re-opened.
- BUG ID 41240 - The **Export Scene Graph Bookmarks** dialog did not prompt before overwriting files.
- BUG ID 41584 - Errors in registering plug-ins for the Gaffer node caused Katana to fail to load.
- BUG ID 42132 - The Viewer required a geometry.knots attribute when drawing curves, even though that attribute and its contents were neither needed nor used.
- BUG ID 42201 - An exception was raised when middle-click dragging a cell in the **Color** column in a row of a light of a Gaffer node that was edited in the **Parameters** tab.
- BUG ID 42463 - The Importomatic CastingSheet.py example's XML tag was named incorrectly. The correct form is <castingsheet>.
- BUG ID 42698 - The functions **pystring.isupper()** and **pystring.islower()** only returned **true** if the input string was set so that every character was of the correct case. Now these functions return **true** if every *cased* character in the input string is of the correct case. This new behavior matches Python's definition of these functions.
- BUG ID 43632 - Asset IDs in Viewer proxies were not being resolved.
- BUG ID 43449 - When certain text fields had focus, keyboard shortcuts such as **Ctrl**+**O** and **Ctrl**+**N**, did not work.
- BUG ID 43467 - Attempting to get the default value of an array parameter through a GenericAssignParameterPolicy instance returned a **None** value.
- BUG ID 43526 - PRMan coshader parameters with the coshaderPort hint enabled did not correctly appear in the connectable list.

- BUG ID 43580 - The 'hide' hint specified in an args file was ignored for coshader ports on shading nodes.
- BUG ID 44238 - The **SocketIdSenderException** raised in the FnRender API by **SocketIdSender** wasn't a subclass of **std::exception**, and didn't provide any description of an error when thrown.
- BUG ID 44257 - PRMan Live Rendering: The render camera occasionally wasn’t set correctly on render start, causing camera updates to be ignored.
- BUG ID 44328 - An editable PopupFormWidget, when clicked in certain regions of the widget, received keyboard focus but not keyboard input, even after additional mouse clicks.
- BUG ID 43525 - In shading nodes, if a coshader parameter enables the coshaderPort hint and has a label hint defined, the label is now used in the connectable port list instead of the parameter name.

  This is only applied when a **nodeType** is set or changed. Existing nodes still show the parameter name.
- BUG ID 44852 - An **includeInCameraList** parameter incorrectly appeared in the parameter interface of LightCreate nodes. The parameter is now hidden.
- BUG ID 19555 - When changing the **application**>**fontSize** preference, certain UI elements were updated only when restarting Katana, notably:
  - titles of groups of parameters, attributes, and project settings,
  - headings of table widgets,
  - annotations in the **Viewer** tab, and
  - text and widgets in the timeline.
- BUG ID 31257 - The keyboard shortcut for the **Edit** > **Dim Nodes Unconnected To Viewed Node** command in the **Node Graph** tab was incorrectly shown as **Alt**+**D** instead of **Alt**+**.** (period).

  The keyboard shortcut for the **Edit** > **Toggle Ignore State of Selected Nodes** command in the **Node Graph** tab was incorrectly shown as **D** instead of **Alt**+**D**. The **D** key toggles the ignore state of a single node under the pointer.
- BUG ID 31890 - The build instructions in the example plug-in READMEs were out of date. These have now been clarified and expanded.
- BUG ID 32778 - The new modifier keyboard shortcuts that were introduced for the commands in the **Render** main menu (see BUG ID 47318 for related information) now allow you to start a Preview or Live Render, even when interacting with the **Viewer** tab, where the **P** keyboard shortcut triggers the **Pin Manipulators** command from the tab's **Manipulators** menu.
- BUG ID 38133 - The colors of icons for items in list and table widgets were multiplied with the selection color, negatively affecting their appearance and readability.
- BUG ID 40154 - The help text for the **nodegraph**>**stickyDrag** preference has been corrected and expanded.
- BUG ID 40611 - A renderer plug-in’s destructor was not called on render completion.
- BUG ID 42446 - When the pointer was over the **Monitor** tab and certain keyboard shortcuts of application-wide actions were pressed, such as **Ctrl**+**O** to open a Katana project, tab-specific actions were triggered instead, such as turning an overlay image on or off (keyboard shortcut: **O**).
- BUG ID 42687 - When opening any of the **Find** pop-ups on the **Project Settings**, **Parameters,** or **Attributes** tabs, hidden project settings, parameters, or attributes were wrongly listed in the popups.

The **Find Parameters** buttons of SuperTools that are edited in the **Parameters** tab are now disabled (made unavailable), as SuperTools provide their own custom parameter interfaces, which may not correspond to parameters on the respective SuperTool node.

- BUG ID 42721 - When reloading a LiveGroup node containing nested child LiveGroup nodes, the nested LiveGroups were unnecessarily reloaded multiple times.
- BUG ID 43089 - **SocketConnection.cpp** and **SocketIdSender.cpp** of the FnRender plug-in API were not including required header files, notably the header file that provides access to the POSIX operating system API, and header files for system socket handling.
- BUG ID 44596 - Previously, Katana's preference file would be written to disk on startup and on each change of preference settings.

  The file is now only written when the **Preferences** dialog is closed using the **OK** button, and on shutdown of Katana.
- BUG ID 45488 - When the **Dim Nodes Unconnected to Viewed Node** feature of the **Node Graph** tab was enabled, connected nodes inside a Group, LiveGroup, or SuperTool-based node were incorrectly dimmed.
- BUG ID 45675 - Double-clicking on the expand/collapse arrows in the **Interactive Render Filters** pop-up widget raised an exception.
- BUG ID 45684 - The **Catalog** tab, when containing a large number of items, was slow to refresh upon switching view or deleting items.
- BUG ID 45861 - Adding multiple coshader terminal ports wasn't supported for NetworkMaterial nodes.
- BUG ID 45916 - When clicking the **Find** button in the parameter interface of a MaterialStack node, an exception was raised.
- BUG ID 45962 - The **Node Graph** tab's **Find** pop-up incorrectly listed deleted nodes.
- BUG ID 45963 - When using animated viewer proxies and changing the current frame, and collapsing or expanding scene graph locations in the **Scene Graph** tab, wrong geometry could be displayed in **Viewer** tabs due to caching issues.
- BUG ID 46020 - If an error was encountered while baking out a Look File to disk, the **Progress** dialog persisted and could not be cleared.
- BUG ID 46624 - When setting node attributes using **NodegraphAPI.Node.setAttributes()**, previously cached node shape attributes were not reset properly.
- BUG ID 46629 - The **source** parameter of a LiveGroup node whose parent node was locked did not appear as locked when the LiveGroup node's parameters were first shown in a **Parameters** tab.
- BUG ID 46382 - When publishing nested LiveGroup nodes to assets, the LiveGroup state was erroneously written to LiveGroup files, leading to warnings and issues when loading them into a Katana project later.
- BUG ID 46754 - When changing the contents of a nested LiveGroup node after making it editable following a revert, **SystemError** and **KeyError** exceptions were raised.
- BUG ID 46827 - Arnold 4.1 screen windows with a height greater than their width were incorrectly processed, resulting in an oversized screen window.
- BUG ID 46838 - When using negative or zero values for the **Slider Minimum**, **Slider Center**, or **Slider Maximum**, widget options of a number parameter, and in certain other configurations of those options, Katana became unresponsive or crashed.

- BUG ID 47098 - When clicking the checkmark icon of the **New Overrides Active For All Passes** checkbox in the **active for passes** pop-up, the LookFileManager node was not updated. Only clicking the label of the checkbox had the desired effect.
- BUG ID 47194 - Input ports for coshader port parameters on ShadingNode nodes weren't hidden when the containing page was hidden.
- BUG ID 47597 - The use of the pixel probe in the **Monitor** tab could, in rare cases, lead to unpredictable behavior, such as graphical corruption in Katana's UI.
- BUG ID 47994 - Uncommitted edits to multi-line text parameters were lost on minimizing the Katana window or switching workspace.
- BUG ID 48051 - When converting a Group node without parameters and without children, but with input or output ports, to a LiveGroup node, the LiveGroup node was not marked as modified.
- BUG ID 48066 - When calling **EventModule.IsCollapsedRegistered()** with a combination of event type and event ID for which no collapsed event handlers had been registered, an exception was raised.
- BUG ID 48188 - When choosing the **Save As...** or **Export Selection...** commands from the **File** menu, messages printed to the console did not take changes to the filename or asset ID made by the asset plug-in into account, such as the addition of the **.katana** file extension.
- BUG ID 48439 - InteractiveRenderFilters and RenderFilter nodes were not shown with the right node type icons in the **Node Graph** tab.