Release Notes for Nuke 6.0v1

This is a major new release of Nuke with many new features, improvements and bug fixes.

6.0v1

Version

Nuke 6.0v1

Release Date

22 January 2010

Supported Operating Systems

- Mac OS X 10.5 "Leopard" and 10.6 "Snow Leopard" (32-bit only)
- Windows XP SP2, XP64
- Linux CentOS 4.5 (32- and 64-bit)

New Features

RotoPaint

The new RotoPaint node (**Draw** > **RotoPaint**) supports multiple shapes and paint strokes, with a non-destructive layer approach and intermediate result caching (to memory and disk) for efficient interactive painting.

You can draw Bezier and B-Spline shapes with individual and layer group attributes, including per-point and global feather, motion blur, blending modes and individual or hierarchical 2D transformations. Paint includes color and effects brushes along with a recursive cloning tool to source any input or frame selection offset with a full 2D source transformation and onion skin preview. Shapes and paint strokes can apply tracker data and expressions and allow point animations to be accessed inside of Nuke's Curve Editor and via Python scripting.

You can choose to set one of RotoPaint's tools as your default tool when opening the RotoPaint node. Select your default tool from the following list and add the Python line in your menu.py:

- Select All
  nuke.knobDefault('RotoPaint.toolbox','0.0')
- Select Curves
  nuke.knobDefault('RotoPaint.toolbox','1.0')
- Select Points
  nuke.knobDefault('RotoPaint.toolbox','2.0')
- Select Feather Points
  nuke.knobDefault('RotoPaint.toolbox','3.0')
- Add Points
  nuke.knobDefault('RotoPaint.toolbox','8.0')
- Remove Points
  nuke.knobDefault('RotoPaint.toolbox','9.0')
- Cusp Points
  nuke.knobDefault('RotoPaint.toolbox','10.0')
- Curve Points

nuke.knobDefault('RotoPaint.toolbox','11.0')

- Remove Feather
  nuke.knobDefault('RotoPaint.toolbox','12.0')
- Open/Close Curve
  nuke.knobDefault('RotoPaint.toolbox','13.0')
- Bezier
  nuke.knobDefault('RotoPaint.toolbox','4.0')
- B-Spline
  nuke.knobDefault('RotoPaint.toolbox','5.0')
- Ellipse
  nuke.knobDefault('RotoPaint.toolbox','6.0')
- Rectangle
  nuke.knobDefault('RotoPaint.toolbox','7.0')
- Brush
  nuke.knobDefault('RotoPaint.toolbox','14.0')
- Eraser
  nuke.knobDefault('RotoPaint.toolbox','15.0')
- Clone
  nuke.knobDefault('RotoPaint.toolbox','16.0')
- Reveal
  nuke.knobDefault('RotoPaint.toolbox','17.0')
- Dodge
  nuke.knobDefault('RotoPaint.toolbox','18.0')
- Burn
  nuke.knobDefault('RotoPaint.toolbox','19.0')

For more information on RotoPaint, see the RotoPaint chapter on page 427 of the Nuke User Guide.

The pre-6.0 Bezier and Paint nodes have been deprecated in favor of the new RotoPaint node. However, Bezier and Paint are still in the application for backwards-compatibility with old scripts. Like any nodes, they can be created via Python and should you find the need (or just feel nostalgic) you can add them to your tool bar menu with a statement in your menu.py file like the following:

- For Bezier:

```
tb = nuke.toolbar("Nodes")
tb.addCommand("Draw/Bezier",
"nuke.createNode(\"Bezier\")", icon="Bezier.png")
```

- For Paint:

```
tb = nuke.toolbar("Nodes")
tb.addCommand("Draw/Paint", "nuke.createNode(\"Paint\")",
icon="Paint.png")
```

Keylight

Keylight (**Keyer** > **Keylight**) is an industry-proven blue/green screen keyer. It is simple to use and is particularly good at tackling reflections, semi-transparent areas and hair. Because spill suppression is built-in, often selecting the screen color is all you need to do to pull the matte and seat the foreground into the background so it looks photographed rather than composited.

Keylight comes with a suite of tools to erode, soften, despot and otherwise manipulate the matte should the need arise. Inside and outside masks are supported. Separate color correction, suppression and edge correction tools are also included to fine tune the result.

For more information, see the Keylight chapter on page 397 of the Nuke User Guide.

NukeX

NukeX is a brand new and feature extended version of Nuke. NukeX gets installed with Nuke 6.0. If you already have Nuke 6.0 installed and have obtained a license for both Nuke and NukeX, you automatically have access to the NukeX features, including FurnaceCore. NukeX includes all the features you have in Nuke, plus the following:

- **LensDistortion**: The new LensDistortion node (**Transform > LensDistortion**) gives you an improved radial distortion model for applying or removing lens distortion. The node features three new methods for automatically analysing lens distortion:
  - **Image analysis** - Tracks features through an input sequence and uses them to calculate the lens distortion.
  - **Grid analysis** - If you shoot using a grid, this type of analysis detects the grid lines and uses them to calculate the lens distortion for greater accuracy.
  - **Line analysis** - If all else fails, you can also use the interface to draw lines along features in the input that should be straight, then calculate the distortion from those.

  A displacement map output also allows you to use Nuke's STMap node to apply the distortion calculated on one image to another.

  For more information, see the LensDistortion chapter on page 909 of the Nuke User Guide.

- **CameraTracker**: With the CameraTracker node (**3D** > **CameraTracker**), you can track the motion in the input 2D footage to create a 3D camera. You can automatically track features or add manual tracks from a Tracker node. You can mask out moving objects using a luminance or channel mask generated from any source and and edit your tracks manually. You can also solve the position of several types of cameras as well as solve stereo sequences. CameraTracker will automatically create a scene linked to the solve containing a 3D camera and point cloud.

  You can also calculate lens distortion to undistort the source on a Card node within the CameraTracker node, without having to open a separate LensDistortion node. On CameraTracker's **Lens** tab, you can choose between **No Lens Distortion**, **Known Distortion**, **Refine Known Distortion**, **Unknown Lens** to account for lens distortion in your solve.

  For more information, see the CameraTracker chapter on page 887 of the Nuke User Guide.
- **DepthGenerator**: The new DepthGenerator node (**3D** > **DepthGenerator**) provides a method to produce a per-frame Z-depth map from the input 2D footage. It additionally requires a camera solve which can be obtained using the CameraTracker node (**3D** > **CameraTracker**).

  For more information, see the DepthGenerator chapter on page 918 of the Nuke User Guide.
- **FurnaceCore**: This plug-in bundle consists of twelve of The Foundry's best Furnace tools including Kronos, the optical flow re-timer, grain and de-grain tools, rig-removal, and more. For more information, select **Help** > **Documentation** in Nuke and have a look at the FurnaceCore User Guide.

For more information on NukeX, see page 884 of the Nuke User Guide.

Feature Enhancements

Python

- Added Python bindings for Curve library and RotoPaint. You can use the following examples:
  - To print out the names of top-level items:

```
k = nuke.toNode('RotoPaint1')['curves']
for thing in k.rootLayer:
  print thing.name
```

  - To calculate how much an object will be translated by at a particular time:

```
knob = nuke.toNode('RotoPaint1')['curves']
transform = knob.rootLayer[0].getTransform()
translationXCurve = transform.getTranslationAnimCurve(0)
translationYCurve = transform.getTranslationAnimCurve(1)
time = 2.0
x = translationXCurve.evaluate(time)
y = translationYCurve.evaluate(time)
print "translated by { %f, %f } at time %f" % (x, y, time)
```

- PythonGeo is a new node that can be used in Nuke, and it provides a Python interface to 3D geometry data. The PythonGeo code can be found in the NDK examples.
- Python scripting documentation has been expanded.

- It's now possible for user-written knobs to have custom user-provided Python bindings as well.
- Nuke's internal vector math classes ('Matrix3', 'Matrix4', 'Quaternion', 'Vector2', 'Vector3', 'Vector4') are now available from the "nuke.math" module.
- Added Python Undo object to Python documentation.
- Added GroupKnob Python bindings.
- Added documentation for the Menu, Menu Item, Menu Bar and ToolBar classes. To see the documentation, select **Help** > **Documentation** in Nuke and click **Python scripting** > **Classes**.
- Previously, Pulldown_Knob.setValues() had no order control. Now it's possible to call Pulldown_Knob.setValues() with a Python sequence to enforce an order on the item list. For example:

```
shots = ['010_100','010_500','010_340','010_200']
shots.sort()
pNode = nuke.toNode('POP')
shotKnob = nuke.Pulldown_Knob("shotSelector","Load Shot")
shotList = []
for shot in shots:
    shotOp = "pNode.loadShot('%s')"  % (shot)
    shotList.append( (shot, shotOp) )
shotKnob.setValues(shotList)
pNode.addKnob(shotKnob)
```

- A means to catch data dropped onto the node graph through Python has been added. The new function nukescripts.dropData is called when the drop occurs. For example:

```
def dropData( mimeType, text ):
  print str(mimeType)
  print str(text)
```

```
return None
```

- You can now use nuke.env["LINUX"]. This returns True if you are running Nuke on Linux, and False if you are not.

NDK

- Several modifications to the 3D API have been made to make it easier to make 3D plug-ins:
  - Moved the following protected methods to the public section because they are query methods that should be accessible to other nodes:

    ```
    virtual CameraOp* render_camera(int sample=0);
    virtual GeoOp* render_geo(int sample=0) = 0;
    virtual Matrix4 camera_matrix(int sample=0);
    virtual Matrix4 projection_matrix(int sample=0);
    virtual void format_matrix(int sample=0);
    Matrix4 get_format_matrix(float cx, float cy);
    ```

  - Added a new method to make Primitive access easier.

    ```
    const Primitive* primitive(unsigned n) const;
    ```

  - Added a virtual method, that gets called right after create_geometry() is called inside the geometry_engine() method. In this way, the SourceGeo plug-in can over-write things like matrix.

    ```
    virtual void init_geoinfo_parms(Scene&, GeometryList&)
    ```

- Added a prefix to Filter knob. You can now add a prefix to a filter knob name to allow you to have multiple filter knobs in one knob.

  Current:

  ```
  Filter::knobs(Knob_Callback)
  ```

  New:

Filter::knobs(Knob_Callback, name="filter", label=0)

- Added a GPU example 'ChannelSelector.cpp' to the NDK.
- NDK: New mechanism to check whether the use of interactive licenses has been forced in render mode.

  Now in DDImage/Application.h:

  bool DD::Image::Application::UsingGUILicense()

  Returns true if in GUI or -i mode.

Preferences

- There is a new **handle pick size** setting on the Viewers tab of the Preferences. This allows you to change the size of the pickable area around the Viewer's control handles (such as points on RotoPaint strokes/shapes), making it easier to select points. By default, it is set to the same size as the **handle size**. You can find the preference in the **Preference** menu, on the **Viewers** tab.
- Added RotoPaint line drawing preferences (**Preferences** > **Viewers** tab) for line width and no shadow.
- Added preferences for changing the color of RotoPaint points, curves, and transform jack. You can find these on the **Viewers** tab of the Preferences.
- Added a new **start file browser from most recently used directory** preference to the **Windows** tab of the Preferences. When this is checked, the File Browser starts from the most recently used directory between sessions. Unchecking this takes you back to the 5.1 style of working where Nuke uses the current working directory.
- On the **Viewers** tab of the Preferences, there's now a new setting called **new Viewers go to own window**. When this is checked, each new Viewer you create appears in its own

floating window. When this is NOT checked, additional Viewers will attempt to dock with existing Viewers.

Documentation

- You can now select **Help** > **Release Notes** in Nuke to go to the release notes web page.
- The User Guide has a new template with an improved layout.

Other Enhancements

- On Linux, you can now press the Windows key and drag to pan the Curve Editor.
- Script Editor highlighting for int, float, True and False.
- The Curve Editor right-click menu now includes a new checkable **Edit** > **Frame Snap** menu item. It is checked on by default and only enabled for the Curve Editor (it doesn't make sense in LUTs or HueCorrect). When activated, the x position of any point moved will be rounded to the nearest integer. When deactivated, the move acts like it did before. Press **Shift** while dragging to momentarily disable the snapping.
- You can now lock a Viewer to a specified zoom level for all inputs. Right-click on the Viewer, and select **Prevent Auto Zoom** to toggle between maintaining the same zoom level for all inputs (on) and changing it according to on-screen image dimensions (off). Alternatively, you can also press **Alt+Z** on Viewer, or toggle **prevent auto zoom** in the Viewer settings.
- ScanlineRender: ScanlineRender node has a new **overscan** control. You can use this control to set the maximum additional pixels to render beyond the left/right and top/bottom of the frame. Rendering pixels beyond the edges

of the frame can be useful if subsequent nodes need to have access outside the frame. For example, a Blur node down the node tree may produce better results around the edges of the frame if overscan is used. Similarly, a subsequent LensDistortion node may require the use of overscan.

- A separate OFX cache has been created for 32-bit and 64-bit plug-ins.
- Nuke now exports TIFF headers correctly. The following headers have been added when writing TIFFs:

  282 - "XResolution"

  283 - "YResolution"

  296 - "ResolutionUnit"

  278 - "RowsPerStrip"

Bug Fixes

- BUG ID 3654 - Undo now works properly when dragging a node between another node and the Viewer.
- BUG ID 3904 - Increased precision in float values when exporting knobs to ASCII.
- BUG ID 4282 - NUKE_PATH environment variable now supports the ';' character as a path separator on Windows.
- BUG ID 4425 - Read node: some files were not recognized as sequences.
- BUG ID 4441 - Autocrop: You can pass arguments for the first, last, increment and layer to analyze.
- BUG ID 5170 - Fixed a problem with loading a QuickTime movie made with AJA Codec v4.0.

- BUG ID 6123 – Added the unsupported FFT, InvertFFT, and FFTMultiply nodes to the **Other** > **All plugins** > **Update** menu.
- BUG ID 6168 – When the Node Graph was active but no node was selected, hitting any of the number keys did not switch the input on the current Viewer, but produced an error.
- BUG ID 6756 – Slashes were appearing as backslashes in some menu items.
- BUG ID 7187 – When precomping multiple selected nodes, the nodes were left disconnected if an error occurred while writing the precomp script out.
- BUG ID 7504 – Apple Pro Res movie file was reading in with an incorrect range.
- BUG ID 7579 – Mac OS X: Script name didn't display in the **Alt**+**Tab** title bar.
- BUG ID 7598 – Setting a key on Card with many vertices hung Nuke.
- BUG ID 7698 – Python: INVISIBLE flag was lost on save (setVisible state).
- BUG ID 7729 – Nuke crashed with the AppleMyungjo (Myungjo) and Gungseouche fonts in the File Browser.
- BUG ID 7792 – Python: The main thread that initializes and executes pyQt widgets didn't catch exceptions.
- BUG ID 7850 – Python: The autolabel() command is now being run when adding a node or when changing channel selection.
- BUG ID 7877 – Nuke now respects FrameCycler range increments.
- BUG ID 7896 – Mac OS X: The size of the Nuke dock icon was reduced to get the correct reflection.

- BUG ID 8103 - Switching between left and right views could fail if importing footage before setting up stereo views.
- BUG ID 8179 - Python help for nuke.menu was missing 'Nodes' and 'Panes'.
- BUG ID 8231 - Fixed a crash caused by position-related Python methods on the root node.
- BUG ID 8249 - ColorCorrect was not drawing color sample lines on Range curves.
- BUG ID 8251 - Mac OS X: Added high-resolution icons.
- BUG ID 8255 - IBKGizmo parameters were disabled after the first change.
- BUG ID 8260 - Fixed Nuke render dialog not supporting negative range rendering.
- BUG ID 8525 - IBKGizmo lost undo functionality after reload.
- BUG ID 8544 - Precomp: Version errors prevented the output node from being created.
- BUG ID 8566 - Fixed **set to this frame** rounding the last number if the frame range was above one million.
- BUG ID 8569 - Fixed the loading of exif metadata in JPEG files.
- BUG ID 8585 - Creating an extra control point would always move the point in Y immediately.
- BUG ID 8603 - .Pic file writer was broken.
- BUG ID 8613 - Removed an obsolete div line in GenerateLUT.
- BUG ID 8614 - Made the **Render** > **Flipbook** process return to the current frame in the Viewer, rather than moving to the last one.

- BUG ID 8635 – Read from Write (including Precomp): stereo exr files/multiple views were not read back in correctly.
- BUG ID 8664 – Viewer Color Sampler was trying to sample regions outside of the bounding box but inside the format (refresh button went red and CPU usage went up).
- BUG ID 8712 – Fixed 3D objects not getting rendered properly in the Viewer when disabling, enabling or selecting them.
- BUG ID 8739 – Progress bars: Granularity was too coarse for multiple Writes.
- BUG ID 8741 – The **Edit** > **Select Connected Nodes** shortcut (**Ctrl**+**Alt**+**A**) was not working on some Windows XP machines.
- BUG ID 8759 – Viewer Color Sampler was showing zero for very small values, rather than using exponential notation.
- BUG ID 8801 – Python: Save script on starting Nuke did not update to saved name.
- BUG ID 8849 – Added missing Panel class documentation.
- BUG ID 8917 – .Pic files were being given an incorrect aspect ratio when read.
- BUG ID 8932 – Provided a Python function 'nuke.runIn()' to execute Python script in the context of a node.
- BUG ID 8935 – Vectorblur crashed Nuke when **grow bbox** was changed and **method** was set to **forward**.
- BUG ID 8947 – ColorTransfer crashed when there weren't enough inputs.
- BUG ID 9002 – Fixed a crash that was caused by calling setValue on a ColorKnob.

- BUG ID 9004 - Gizmos within gizmos lost settings when the group was copied and pasted.
- BUG ID 9009 - Glow: There was a crash when **non linear** was enabled.
- BUG ID 9028 - Multiline_Eval_String_Knob became a normal String_Knob after save and reopen.
- BUG ID 9052 - OK and Cancel buttons from showModalDialog() did not show in the right place.
- BUG ID 9061 - Nuke was writing frame rates in a way not understood by Final Cut Pro on 25fps and 59.94 fps mov files.
- BUG ID 9093 - Fixed a crash happening with a Python panel.
- BUG ID 9120 - Python: Setting incorrect values in a format control crashed Nuke.
- BUG ID 9151 - Fixed fbx_node_name getting changed through a Python script.
- BUG ID 9161 - Python: The nuke.layers() function has changed. In previous versions it was returning a list of lists of strings, some of which would be empty. It now just returns a flat list of strings. If you have any scripts which use nuke.layers(), you will need to adjust them accordingly.
- BUG ID 9191 - Backdrop node: Auto-backdrop now constrains the size based on the selection.
- BUG ID 9206 - Nuke.filename(None) crashed Nuke.
- BUG ID 9222 - Changed the init.py and menu.py that are shipped with Nuke so that their calls to pluginAddPath use addToSysPath=False.
- BUG ID 9227 - There were tooltip errors on Flare node's **position** and **radius** controls.
- BUG ID 9262 - Re-enabled Sapphire plug-ins.

- BUG ID 9266 - The Transform node scale handle now moves at the same rate as the cursor.
- BUG ID 9282 - Documentation: Added the lack of command line render to PLE restrictions.
- BUG ID 9365 - An issue with Nuke removing nodes below a bad gizmo was fixed.
- BUG ID 9369 - Python: Setting Int_Knob to none caused a crash.
- BUG ID 9388 - Gaps were appearing in stereo properties in a customer script.
- BUG ID 9412 - Corrected line indentation of the 'filenameFix' example in the documentation.
- BUG ID 9414 - Gizmos: Nuke crashed when editing missing knobs.
- BUG ID 9424 - Fixed a Python panel crash.
- BUG ID 9446 - Color control drag and drop clamped color values in the range 0 to1.
- BUG ID 9470 - Segmentation fault when opening a bad EXR file.
- BUG ID 9482 - Clicking OK in a Python panel crashed Nuke.
- BUG ID 9483 - Fixed a crash in the TIFF file reader.
- BUG ID 9487 - Curve Editor: Selecting **Predefined** > **Loop** in the right-click menu and canceling it still applied loop.
- BUG ID 9524 - Nuke crashed when reading images of height much bigger than width.
- BUG ID 9533 - GPUOp did not refresh when the knob was changed.
- BUG ID 9540 - VectorField: Allowed importing of .csp LUT files with no metadata.

- BUG ID 9556 - Fixed a problem with SplineWarp keys not appearing in the timeline.
- BUG ID 9586 - Using **create keyframes** in the Reconcile3D node controls crashed Nuke occasionally.
- BUG ID 9616 - Added nuke.addFavoriteDir in the Python Help.
- BUG ID 9626 - Nuke crashed when a timecode was added to a malformed DPX file using an AddTimeCode node before a ViewMetaData node.
- BUG ID 9678 - A customer script crashed when using the DiskCache precache.
- BUG ID 9684 - Stereo: A split off left state was not recorded in a Nuke script.
- BUG ID 9775 - Exif data was corrupted when passing a JPEG through Nuke.
- BUG ID 9800 -
- BUG ID 9823 - Added checkboxes to some Viewer menu items.
- BUG ID 9858 - Fixed the .r3d reader tooltip to suggest leaving gamma set to linear.
- BUG ID 9863 - Fixed the Flipbook default frame range.

Known Issues and Workarounds

RotoPaint

- The foreground onion skin overlay updates as you paint. This will change so the overlay only updates with the new stroke on pen up.
- Python: The RotoPaint Python API has been restricted to read-only.

- BUG ID 9199 - Painting is slower with other nodes' (such as a Read node) control panels open. Paint is the fastest when all control panels are closed except for the RotoPaint panel.
- BUG ID 9238 - Painting on Mac OS X and Linux is slower when the paint cursor is near the edges of the screen.
- BUG ID 9782 - Drawing slows down when multiple layers have been created and motion blur is enabled.
- BUG ID 10024 - There is a known issue with expression linking, whereby **Ctrl**/**Cmd**+dragging from one Animation menu to another Animation menu to set up expression links, will result in incorrect values being transferred. Workaround: **Cmd**/**Ctrl**+dragging each knob value separately will result in the correct values being transferred.
- BUG ID 10050 - Cloned RotoPaint nodes currently cause Nuke to crash.

Other Known Issues

- Documentation: Several bookmarks in the user guide pdf that gets installed with Nuke do not appear at the right level. This will be fixed for the next release. In the meantime, you can use the table of contents, the index, or the search function to find what you're looking for, or download a corrected user guide pdf from our website at www.thefoundry.co.uk (from the Nuke product page, select **User Guide**).
- On 32-bit Windows XP, writing QuickTime files to UNC paths may not work if you are using an older version of QuickTime. This is due to a bug in QuickTime rather than

Nuke. The solution is to use the latest version of QuickTime.

- QuickTime is not provided by Apple for Windows 64-bit applications and is not available in the Windows 64-bit version of Nuke at this time.
- On Mac OS X 10.5 (Leopard), when the Viewer is set to the **OpenGL stereo** stereo display mode, Nuke may trigger an OS X bug that causes a kernel failure. This is due to a bug in OS X 10.5 to do with stereo OpenGL support. For this reason, we do not recommend using the **OpenGL stereo** stereo viewing mode in Nuke on Leopard at this time. The bug has been registered with Apple as bug number 5897735.
- We direct FrameCycler to write to the user's Nuke temp directory (NUKE_TEMP_DIR) for its user settings files. You can redirect this by modifying the FrameCycler/settings/Global_Settings.xml file that can be found within your Nuke installation.
- If you have trouble with FBX files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk web site. It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac OS X, and Linux.

  To download the FBX converter:

  1. Go to http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=10775855.

  2. Scroll down to **FBX Converter** and click on one of the links to start the download.

- There is a Python syntax conflict when assigning knob names on the fly with **nuke.nodes.<node>()** if the knob is called 'in'.

  For example, this will give a syntax error:

  nuke.nodes.Shuffle(in = 'depth')

  while this works because 'in' is a string here and not a keyword:

  sh = nuke.nodes.Shuffle()

  sh['in'].setValue('depth')
- BUG ID 5063 - ScanlineRender: orthographic projection mode not working. This was fixed earlier, but the fix caused bug 5978 and so has been removed. The bug will be addressed more correctly in a subsequent release.
- BUG ID 5083 - Flipbooking the output of the Anaglyph node asks which view you want to render. This question is unnecessary as the result is an anaglyph image. Irrespective of what view you choose, the flipbook output will be the same.
- BUG ID 5690 - Windows run-time libraries were not packaged properly with Nuke.

  Nuke will now run correctly from a network install on Windows without specifically installing the run-time libraries, though we still recommend that you do so as there will still be some minor problems without them. For details, please see *Installation on Windows* in the *Installation and Licensing* chapter of the user guide.
- BUG ID 5922 - At the moment, cloning does not work properly with all OFX nodes. This affects, but is not restricted to, any nodes that have an analysis pass.
- BUG ID 6455 - You should not call the Python command nuke.restoreWindowLayout() from the Script Editor as that

can cause Nuke to crash. Instead, you can use the same command from your menu.py, restore layouts by selecting **Layout** > **Restore Layout**, or use a custom menu or toolbar item.

- BUG ID 6896 - On Linux, UI corruption may occur if you are running Nuke under window managers that support OpenGL-based effects (for example, Compiz or Beryl) and the effects are turned on (that is, **System** > **Preferences** > **Appearance** > **Visual Effects** has been set to either **Normal** or **Extra**). The solution is to set **Visual Effects** to **None**.
- BUG ID 8063 - Creating many new nodes with *nuke.createNode()* and the inpanel argument at default (True) may crash when too many node control panels are created too quickly. The workaround is to pass the inpanel argument as False or else use *nuke.nodes.NodeClass()* (where NodeClass is the type of node to create) to create the node and then connect it to the currently selected node manually.
- BUG ID 9521 - Currently, the Nuke Viewer cannot cache very large plate sequences in float. The limit per frame is 50MB. If your frames are larger than this, you may need to switch to proxy mode for the caching to work.
- BUG ID 10048 - With some larger scripts, playback performance of cached frames will be improved by turning off the Viewer overlay. This currently means that Viewer playback performance may be faster when turning off the overlays in the Viewer, by pressing 'O'.
- BUG ID 10056 - Image not caching correctly with a LensDistortion node, when advancing in the timeline. If you apply a LensDistortion node to an input, zoom in, then jump to another frame and zoom out, as you zoom out the

newly-visible areas of the image don't get filled in properly but remain black. The workaround is either not to switch frames while zoomed in, or to clear the disk cache in order to restore the missing data (pressing Refresh does not work). The same problem affects the CameraTracker when **Undistort** is switched on under the **Lens** tab.

DEVELOPER NOTES

Here are the changes relevant to developers.

Changes for Nuke 6.0v1

- Plug-ins compiled for Nuke 5.x will require rebuilding for use in Nuke 6.0v1 due to binary format changes. Any plug-in built for 6.0v1 will remain compatible with versions 6.0v2 and so forth. Every effort will be made to preserve binary compatibility between Nuke 6.0 and future minor version updates such as 6.1, but it is possible that recompilation will be needed. Future major version updates, such as 7.0, may require some source code modification as not all changes are foreseeable at this stage.
- For forward compatibility with future versions, some changes have been made to the Op class. The methods Op::abort() and Op::aborted() are now non-static, and so must be called with an Op object. For the majority of plug-ins, this should make no difference, as their calls to these functions will generally be from within other Op methods. Other data changes have been made, but should have no impact on plug-in developers other than requiring a recompile with the new libraries and headers.
- User-written knobs can now have custom user-provided Python bindings too.
  - Knob implementations that wish to make use of this will need to override the pluginPythonKnob() method to

return a suitable value. The easiest way to implement this is probably to make your knob inherit from PythonPlugin_KnobI as well as the base Knob class and have the pluginPythonKnob() method return this.

- The PythonPlugin_KnobI class is simply a way for the knob to give Nuke a Python PyTypeObject* to use for the bindings. It is up to the knob writer to create this object and write all necessary code for the bindings.

- To free up additional flags, the flag BBOX_WH_BUTTON has been removed, without replacement. This was used internally to indicate whether the BBox control was displaying x/y/r/t or x/y/w/h.
- A new virtual function, "build_splits" has been added to Op. This is called during Op generation, and is a suitable place for doing any initialisation work that split_input and inputContext will wish to refer back to. Note that since 5.2 knobs that split_input and inputContext (and now build_splits) use have needed to have the EARLY_STORE flag set on them to be stored by this time.
- New knobs: Toolbar and TabGroup. Toolbars allow you to declare certain knobs to appear in the Viewer, like the RotoPaint toolbar. Within a TabGroup, Tab knobs will appear as grouped sub-tabs rather than as main tabs on the node panel (also see RotoPaint).
- Handles now support cursors. Pass in a cursor constant (from ViewerContext::Cursor) as the last parameter to make_handle or begin_handle.
- Certain classes that should always have been non-copy-constructable and non-assignable have been made so.
- The Filter knob-creating helper now takes a name, so you can have multiple of these in an Op. The similar Shutter

Controls helper now also allows custom names to be specified per-knob.

- A new predefined channel, "Chan_RotoPaint_Mask", has been added.
- BUG ID 7543 - NDK: Made a much faster Op::close() method.
- BUG ID 8229 - NDK: Added a mechanism to check whether Nuke is using an interactive license.
- BUG ID 7866 - Added an additional sampling method to DDImage::Iop that area samples shadowmaps.
- BUG ID 8282 - Replaced the current slow normals derivation algorithm with new faster one.
- BUG ID 9157 - Render.h API requests: Made the following functions virtual so that Render subclasses can better customize their shutter/time sampling behavior:

  virtual int multisamples() const;

  virtual Scene* scene(int n=0) const;

  virtual double shutter() const;

  virtual double offset() const;

  virtual unsigned samples() const;
- BUG ID 9134 - Scene.h API requests: Made the 'validate()' and 'request()' methods (line 91, 92) virtual so that subclasses can add to their functionality.
- BUG ID 9135 - PrimitiveContext.h API requests: Added a virtual destructor method so that subclasses can be constructed off it.
- BUG ID 9136 - LightContext.h API requests: Added a virtual destructor to LightContext and changed all references to this class so that subclasses can be constructed off it.

- BUG ID 9258- Added a simple GPU example plug-in.
- BUG ID 9362 - Convolve.cpp and movReader examples in the NDK did not compile by default.

  The examples now build correctly.
- BUG ID 9474 - DDImage forward compatibility changes.
- BUG ID 9808 - A Visual Studio solution for building plug-ins with the NDK has been added. You can find it in the folder "Documentation\NDK\vc8" under the directory where Nuke is installed. Instructions on using it can be found in the file "readme.txt" in the same folder.
- BUG 9885 - NDK: Implemented node_name().