



RELEASE NOTES FOR NUKE 5.1v1

This is a major new release of Nuke with many new features, improvements and bug fixes. These release notes are organised into two sections, the first describes changes between Nuke 5.1v1 and Nuke 5.0v2, the second is a detailed history of the Nuke 5.1v1 beta releases.

5.1v1

Version

Nuke 5.1v1

Release Date

03 September 2008

Supported Operating Systems

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

New Features

- Support for 64-bit operating systems on Windows and Linux.
- Mac OS X 10.5 (Leopard) is supported.
- The Nuke Personal Learning Edition (PLE) is now available. The PLE is a special version of Nuke you can run without a valid license key. It is aimed at students, industry professionals, and others interested in Nuke. To get the PLE, you need to install Nuke as usual and launch the PLE in a special way. For more information, refer to the *Installation and Licensing* chapter of the user guide.
- 3D:
 - There is a new Light node that can be set to act as a spot light, point light, or directional light. The old Light node that only supported point lights has been relabeled to Point node to distinguish it from the new Light node.
 - You can import cameras, lights, transforms, and meshes into Nuke from .fbx files created in other applications. This option is available on Nuke's Camera, Light, Axis, and ReadGeo nodes. Nuke uses FBX 2009.1.

- The Card node has extended bicubics (bicubics with more control points). They allow you to subdivide a card, giving you finer control for warping an area. You can subdivide the card into an evenly spaced grid, or pick a location to add a row, column, or both.

For more information on the above features, see chapter 12, *3D Compositing*, in the user guide.

- Stereo workflow:
 - The Views > Stereo menu in the Toolbar has a new node called MixViews. The MixViews node displays a blend between two views, allowing you to easily check how elements in these views are aligned. For more information, see *Displaying Views in the Viewer* in the *Working with Stereoscopic Projects* chapter of the User Guide.
 - In the Views > Stereo menu, there is now a ReConverge node. It lets you shift convergence (the inward rotation of the eyes or cameras) so that any selected point in the image appears at screen depth when viewed with 3D glasses.

To change convergence, you need a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in the other view. You can create a disparity field using The Foundry's O_DisparityGenerator plug-in, which is included in the Ocula plug-in set, or a 3D application.

If you have Ocula installed, you can choose to use it when changing convergence. If you select to use Ocula, extra refinements are done when performing the convergence shift to produce more accurate results with disparity fields that are estimated by a software process (like Ocula) rather than generated from a renderer where they are assumed to be exact.

For more information, refer to chapter 13, *Working with Stereoscopic Projects*, in the User Guide.

- In the controls of the Paint node, Bezier node, and any nodes, groups, or gizmos that have controls for x and y coordinates, there is a new Correlate function. It lets you reproduce changes made to one view in the other.

To do so, you need a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in the other view. You can create a disparity field using The Foundry's O_DisparityGenerator plug-in, which is included in the Ocula plug-in set, or a 3D application.

When Ocula is available, you can choose between using Nuke's Correlate function or Ocula's O_Correlate plug-in. If you use Nuke's Correlate function, the changes are reproduced by adding the disparity vectors in the disparity field to the current values. If you use Ocula's O_Correlate, extra refinements are done to improve the results where the disparity map is not perfectly accurate.

For more information, refer to chapter 13, *Working with Stereoscopic Projects*, in the User Guide.

Improvements

- Colour picker:
 - Changed the Colour Picker slider order to TMI HSV RGB A.

- The colour wheel now offers the Nuke 4 style hue wheel. It can also be zoomed and panned like the Node Graph and Viewer.
- You can show or hide the colour wheel by means of a button, rather than by resizing the window. Clicking on the colour wheel button cycles through three states: the colour wheel, the colour square, and hide the colour wheel/square.
- You can Ctrl/Cmd+click on the colour wheel to only affect the hue of the selected colour, and Shift+click to only affect the saturation.
- The colour picker window includes colour swatches that can remember colours. There is a button that lets you show and hide the swatches. To use a colour from the swatches, left-click on a swatch. To save a colour in a swatch, right-click on the swatch.
- Removed the lock icon from Colour Picker to save screen real estate. When you have one Colour Picker window open, you can now Ctrl/Cmd+click on a colour picker button to open another.

For more information, see chapter 2, *Using the Interface*, in the Getting Started section of the user guide.

- 3D:
 - Geometry objects now have their own transform controls. You no longer need to add a TransformGeo node after the object to manipulate its position, scale, and rotation in 3D space.
 - The Spotlight node now has a “cone penumbra angle” parameter, which controls the softness along the edge of the area of illumination. This feature is only visible in the rendered objects and not in the 3D OpenGL viewer.
 - The Spotlight and Point nodes’ “falloff” toggle has been replaced with a “falloff type” pulldown menu. It controls how much light the object gets from the light source, and is based on the distance between the light source and the object. A Linear type will diminish the light at a fixed rate as it travels from the object. Quadratic and Cubic types will diminish the light at an exponential rate.

For more information, see chapter 12, *3D Compositing*, in the user guide.

- Stereo workflow:
 - You can set up views for stereoscopic projects simply by clicking the “Set up views for stereo” button on the Views tab of the project settings. This deletes the view called “main” and creates views called “left” and “right”.
 - You can assign colours to all views that exist in your project settings by clicking on the colour bar next to the view name. If you also check “use colours in UI?”, the colours are used in Node Graph connections, split views indicators on nodes, and the controls of certain nodes and the Viewer, making it easier to differentiate between views.
 - You can now split all the controls of a node by right-clicking on the controls and selecting “Split all knobs”.
 - Split controls can be expanded to show the value for each view that exists in the project settings. This allows you to give separate values for each view. To expand a control you have split off, click on the small triangle on the left side of the control.

- Added “views” controls to the Anaglyph node properties panel. These controls let you select which views you want to use for the left and the right eye.
- Removed the “stereo” check box on the .exr writer; it now figures out whether to do this or not based on the filename and views to be rendered.

For more information, see chapter 13, *Working with Stereoscopic Projects*, in the user guide.

- Python:
 - The default sys.path in Nuke now includes NUKE_PATH. Paths should be added in init.py for this change to be reflected in Nuke. Alternatively, you can also use the environment variable before starting Nuke.
 - If you now add knob defaults using the command `nuke.knobDefaults()`, it is recommended that you place them in `init.py` rather than `menu.py`. This ensures that they are set for command line start-up as well as GUI (graphical user interface). If these statements are not processed until `menu.py` (GUI start-up), then nodes created from the command line Nuke Python prompt (i.e. starting as `'nuke -t'`) will not have the defaults applied.
 - There is now an object in the Nuke module called `env` that gives information about the particular Nuke environment the user is running on. It can be accessed in a dictionary form, `nuke.env["key"]`.

You can find out what's in the Nuke "global environment" by entering the following in the Script Editor:

```
print nuke.env
```

Some of the items are read-only (for example, `'64bit'` and `'ple'`), and others can be set (for example, `'threads'`).

- The `nuke.nodes` syntax has been augmented. It can now also take name and input parameters, for example:

```
r1 = nuke.nodes.Read(file = "<file path>")
r2 = nuke.nodes.Read(file = "<file path>")
t = nuke.nodes.Merge(name = "My Merge", inputs = [r2, r1])
```

- Python `removeKnob()` command no longer asks for verification.
- Instead of `nuke.animation(...)`, you should now use the following syntax:

```
#create the node
b = nuke.nodes.Blur()

#get the size knob
k = b['size']

# create an AnimationCurve for the channel 0
k.setAnimated(0)

# access all animation curves for the knob
```

```
a = k.animations()
# set the animation keys
a[0].setKey(0, 11)
a[0].setKey(10, 20)
```

For more information on AnimationCurve and AnimationKey objects, refer to the NDK.

- Simplified Group node bindings.

Before, when you created a node, for example, it would always be created inside the Root node.

If you used `nuke.nodes.Blur()`, the node would show up in the Node Graph.

If you then added `nuke.nodes.Group()`, wanted to create a node inside the Group, and tried `nuke.nodes.Blur()`, the node would still open in the Node Graph window (Root node).

To solve this issue, Group nodes (this includes the Root) are context objects and `nuke.nodes.Blur()` is effectively equivalent to:

```
from _future_ import with_statement
r = nuke.toNode("root")

with r:
    nuke.nodes.Blur()
```

For a group node:

```
from _future_ import with_statement
g = nuke.nodes.Group()

with g:
    nuke.nodes.Blur()
```

- Added a Python `scriptClear()` function to clear out the current script.

For more information on the above improvements, see the NDK as well as chapter 17, *The Script Editor and Python*, in the user guide.

- Preferences:

- Renamed "flipbook cache" preference setting to "disk cache", which describes it better.
- You can have all Python commands executed by yourself or Nuke appear in the output pane of the Script Editor when you check "echo python commands to output window" on the Script Editor tab. For example, if you select a node from the Toolbar, the corresponding Python command is displayed in the output pane.

Note that this does not apply to all actions you take in the GUI, only those that are performed by executing Python script commands.

- You can change the type, weight, angle, and size of the font used in the Script Editor.

For more information, see chapter 15, *Setting Interface Preferences*, in the user guide.

- Node Graph:

- You can now press F on the Node Graph to zoom to selected nodes. As before, if no nodes are selected, pressing F on the Node Graph fits all nodes in the space available.
- The Node Graph is now drawn in OpenGL for improved speed.
- When expanding the Node Graph with the space key, the nodes are now kept in the same place on the screen.
- Masking:
 - By default, the alpha channel from the mask input is used as a matte whenever the node's mask input is connected. You can change the channel by using the "mask" pulldown menu in the node's controls. To use a channel from the main input as the mask, you need to make sure nothing is connected to the node's mask input, and select the channel you want to use from the "mask" pulldown menu. The mask input arrow disappears as it is no longer being used. You can get it back by setting the mask channel to "none". For more information, refer to chapter 2, *Channels*, in the user guide.
 - There is a new predefined channel for masks called mask.a, and a new "inject" option in mask controls. If you check inject, the mask from the mask input is copied into the predefined mask.a channel. This way, you can use the last mask input again downstream. You can also set a stream of nodes to use mask.a as the mask, and then change the masking of all of them by simply connecting a new mask into the mask input connector of the first node. For more information, refer to chapter 2, *Channels*, in the user guide.
- User knobs dialog:
 - You can now delete multiple knobs at once. You can also delete an entire tab by selecting all the items including the tab name.
 - Manage User Knobs now offers a script button that can execute Python code in addition to the existing TCL script button.
- Read and Write nodes:
 - Added support for # in filenames for frame number.
You can now use, for instance:
img#.jpg for img1.jpg
img##.jpg for img01.jpg
img###.jpg for img001.jpg
img####.jpg -> img0001.jpg.
The old way of using, for example, img%04d.jpg for img0001.jpg continues to work, too.
 - When rendering multiple Write nodes, they are now all executed before advancing the frame, rather than rendering each Write all the way through the sequence. This can help to improve caching performance.
 - QuickTime codec options are now exposed on the Write node.
 - Added support for reading alpha channels from and writing them into QuickTime movies. Note that some codecs (such as YUV-based codecs) may not honour this.

- Overhauled .dpx reader/writer to handle many more dpx variations. Support for 16-bit dpx files.
- When an execute command fails, it now traces the error message to the console.
- Added exr support for B44 compression (OpenEXR 1.6.0).
- Merge node:
 - The operation types are now listed in alphabetical order.
 - Channel sources have been moved to the second tab.

These changes make the properties panel more intuitive and easier to use.

- Licensing:
 - Added extra directories to the licence file search path on Windows Vista (C:\ProgramData\The Foundry\FLEXIm) and Mac OS X (/Applications/TheFoundry/FLEXIm, /Library/Application Support/TheFoundry/FLEXIm).
 - License error messages now show the lmhostid number.
- Improved the icons in node properties panels and the Viewer controls. Less frequently used controls in the node properties panel title bar have been moved to the Node tab. Hotkeys and Edit menu entries have been added for the postage stamp and hide input actions.
- Added Revert, Cancel, and Close buttons on floating Control Panel windows. The Revert button reverts any changes made after the control panel was opened. The Cancel button does the same as the Revert button but also closes the control panel. The Close button simply closes the control panel.
- Colour sampling: you can now Ctrl/Cmd+click to sample a colour from the Viewer, and Ctrl/Cmd+Alt+click to sample a colour from the node's input while viewing its output. Primate is a special case and always samples from the input.
- Transform > Tracker node: added an "update viewer" checkbox. When checked, the entire image is redrawn during processing. If unchecked, only the search area is updated.
- You can now apply tracking (and other transform) data to individual Bezier points. To do so, Ctrl/Cmd+drag and drop the track's animation button on the Bezier point.
- VIEWER_INPUTs can now transform modify the image data.
- At user request, turning "reformat" on in the Crop node controls deactivates "black outside" (and vice-versa) since this is the most common use pattern.
- Animating parameters: the properties panel right-click menu now includes a "Set key on all knobs" option that you can use to set a key on all controls of the node at their current value.
- The SphericalTransform MirrorBall now uses a technically correct transform. The old transform maths is now called 180 degree FishEye, and is still available in the node.
- Nodes in the Color > Math menu: you can now split the 'value' control into four fields to edit the values assigned to individual channels.
- Added support for tiled rendering of OFX plug-ins that are capable of it, which often results in significantly lower memory usage.
- Improved interactive performance by caching inputs to nodes open for editing.
- The default Truelight node now uses FilmLight's v3 libraries. A Truelight2 node is also included for backwards compatibility and for customers who only have a v2 license.

You only need to be concerned about this if you are using Truelight profiles that require advanced features made available by an additional license purchased from FilmLight. If you want the Truelight menu item to continue to create the old version of the node (Truelight 2), you will be able to override the node creation by putting the following in your menu.py:

```
nuke.menu("Nodes").addCommand("Color/Truelight", "nuke.createNode(\"Truelight\")", icon="Truelight.png")
```

You can also create a Truelight2 node by pressing X and running the TCL command "Truelight2" (without the quotes).

- Default start menu folder names on Windows now add "(32 bit)" when appropriate to avoid overwriting 64-bit shortcuts.
- OFX plug-ins now pass through channels not being operated upon.
- Fixed version support commands to handle a variable number of digits.
- Fixed an issue with wacom pen driver losing events.
- Increased the LUT size to prevent quantisation in the blacks with the Kodak DLAD.
- Tidied up menu item labels, documentation, and tool tips.

Bug Fixes

- BUG ID 1394 - Grain node should keep the incoming bounding box if set to "Apply only through alpha".
- BUG ID 1418 - The Nuke icon only appears on the first opened script.
- BUG ID 1701 - Resize widget not appearing in the corner of the main Nuke window.
- BUG ID 1811 - Text node's font path defaults to /job/common.
- BUG ID 1937 - Bezier / Draw nodes: fixed inconsistent output when 'invert' is on. The 'clip to' control decides exactly how far the invert fills out to. The default stops at the current bounding box.
- BUG ID 2067 - Time > FrameRange node not taken into account in proxy mode.
- BUG ID 2094 - Text node: knob_default Text.font "/some/path/to/fonts" command fails.
- BUG ID 2351 - Artifacts and smearing in VectorBlur.
- BUG ID 2397 - User Guide: 'Using Practical Grain' is out of date.
- BUG ID 2433 - Configured FrameCycler to write out settings files to Nuke temp directory to make sessions persistent. For more information, see *Known Bugs and Workarounds*. The warning messages about QuickTime libraries not being present should now only appear once.
- BUG ID 2638 - Read QuickTimes: files recorded with high/low quality only playback low in Nuke.
- BUG ID 2639 - Paint produces "a coding error in an Op".
- BUG ID 2738 - Paint: on Windows XP, brush stroke appearance is delayed until mouse stops moving or goes out of the bounding box.
- BUG ID 2855 - Dual screen saved layout on a single screen system messes up your layout.
- BUG ID 2908 - Reconcile3D and PointsTo3D tooltips show formatting marks.
- BUG ID 3011 - Side input mask pipe disappears when its not selected.

- BUG ID 3020 - Console window is not reopened when closing a Nuke script (Windows only).
- BUG ID 3078 - Bezier contextual keyboard shortcuts not working.
- BUG ID 3109 - Rendering to TGAs from a static input produces inconsistent output.
- BUG ID 3161 - Interface resize amount is incorrectly calculated if you're in fullscreen mode and have the drop down OS X menu down.
- BUG ID 3187 - Ramp: inverted Ramp node resets bounding box.
- BUG ID 3188 - Merge: 'bbox' knob breaks bbox expressions.
- BUG ID 3202 - Slower render on Windows with a customer script.
- BUG ID 3283 - When 'tearing' a window panel off to place it elsewhere, there is no overlay drawn.
- BUG ID 3351 - A test Tracker script and a Furnace wire removal script cause Nuke to crash.
- BUG ID 3497 - Bezier contextual keyboard shortcuts not working.
- BUG ID 3498 - Python distribution missing modules on different operating systems.
- BUG ID 3507 - Viewer renders frame, then distorts it (only on OS X PPC).
- BUG ID 3521 - Bezier: delete points on spline crashes Nuke.
- BUG ID 3523 - TransformGeo: zooming out causing geometry to be modified/drawn in a strange fashion.
- BUG ID 3525 - Bezier: undo not working correctly.
- BUG ID 3579 - Linux KDE window manager auto hides windows flagged as 'utility' when Nuke loses focus.
- BUG ID 3680 - The Dilate node deadlocks.
- BUG ID 3709 - Empty values should revert to previous value if focus is moved away rather than remaining empty.
- BUG ID 3777 - Curve Editor: cannot add a point anymore without first clicking on a point.
- BUG ID 3807 - Crash when disabling Normals node and trying to enable one of the 'show vertex' buttons.
- BUG ID 3824 - Check boxes do not highlight correctly on mouse exit after mouse down.
- BUG ID 3865 - Provide option to scale the normals display in the 3D viewer. The icon size setting on the Viewers tab in the Preferences can now be used for this.
- BUG ID 4115 - 3D overlays drawn much larger than previously, making them appear clumsy and difficult to judge exact position.
- BUG ID 4179 - Quit shortcut in a group window exits without saving.
- BUG ID 4226 - The ~ shortcut to show a hidden Curve Editor isn't working.
- BUG ID 4249 - A customer script with a Crop node rendering incorrect result.
- BUG ID 4262 - 'Segmentation fault' error on a customer script.
- BUG ID 4321 - Viewer playback is slower when not moving the mouse.
- BUG ID 4359 - JoinViews node: inputs are set by index; however, when pasting into another script the index to name mapping changes.
- BUG ID 4413 - Frame boundaries/film gate clipped by near plane.
- BUG ID 4416 - Read: a TIF file crashes Linux on Read unless you set threads to zero.

- BUG ID 4419 - Improve caching performance.
- BUG ID 4474 - Viewer text occlusion.
- BUG ID 4478 - Ctrl/Cmd + Shift + X on a node in the Node Graph does not disconnect the node if there is no input.
- BUG ID 4514 - A particular VectorBlur test script crashes Nuke.
- BUG ID 4518 - OFlow hang when rendering from command line.
- BUG ID 4521 - TCL (Python): filename_fix Tcl callback query.
- BUG ID 4524 - Edit > Node > Filename > Version Up/Down used to work (pre 5.0) with upper case 'V'.
- BUG ID 4526 - Missing the option to turn off auto label and auto colour in the Preferences.
- BUG ID 4545 - Import Discreet LUT is broken.
- BUG ID 4557 - Viewer's gamma slider: Y indicator (Toggle gamma between 1.0 and the previous setting) turns red when offset but is not red when saved script opened.
- BUG ID 4558 - Group tabs disappear when changing layout. They should remain next to the Node Graph.
- BUG ID 4566 - Viewer node does not disappear / get deleted when Viewer is closed and preference is set appropriately.
- BUG ID 4570 - Frame expression does not work for .mov files.
- BUG ID 4580 - Nuke flagging premult state wrong to OFX plug-ins.
- BUG ID 4617 - Editing user knobs: 'Start new line' flag not saved in scripts.
- BUG ID 4656 - Python: knob.addDefault doesn't work on Text.font.
- BUG ID 4657 - Write in proxy mode with no proxy file name should error.
- BUG ID 4661 - Colour sampling stops working after first click.
- BUG ID 4671 - Channel selector: on creating a channel, the new channel is not automatically assigned in the channel menu.
- BUG ID 4686 - Pressing 'Auto-Compute' with no inputs connected to Primatte causes it to crash.
- BUG ID 4687 - Keyer widget doesn't stretch when panel is expanded.
- BUG ID 4710 - File > Open: script filter should default to .nk instead of the name of the script already open.
- BUG ID 4750 - Erode (fast): crashes when input is a Bezier with opacity set to 0.
- BUG ID 4756 - Importing Ascii into a newly created root LUT causes Nuke to hang.
- BUG ID 4758 - Black x's and white arrows are hard to read in some Shuffle node boxes when Silver colour preset is selected in the Preferences.
- BUG ID 4818 - A Group exported as gizmo doesn't heed 'Start new line' set in 'Manage User Knobs' edit.
- BUG ID 4819 - Bezier: Crash when editing 'shape' control's animation curve in Curve Editor with Edit > Move.
- BUG ID 4833 - Root name is different in GUI and terminal mode.
- BUG ID 4851 - Can't disable (keyboard 'D') a multiple-node selection in Node Graph when a Dot node is included.

- BUG ID 4866 - Saving preferences to a home directory on a UNC path fails with a 'cant create .nuke directory' error message.
- BUG ID 4894 - ShuffleCopy edit channel switched to 'none' causes Nuke to crash.
- BUG ID 4896 - Memory allocation failure in a customer script.
- BUG ID 4908 - Return typed to 'new layer' dialog ignores changes to text field.
- BUG ID 4929 - Add an option to `nuke.show()` to force show as a floating panel.
- BUG ID 4944 - User Guide: Oflow parameter 'Shutter Time' references figure not present.
- BUG ID 4947 - Dilate node on a roto shape causes segmentation fault.
- BUG ID 4950 - MotionBlur in several operators: locking up when you try to Write.
- BUG ID 4955 - 'F' to frame expression only frames between 0 and 1.
- BUG ID 4958 - Can't use Python "threading" module.
- BUG ID 4973 - Python: `nuke.addFormat` should return the created format rather than True or false.
- BUG ID 4989 - Floating panels have redundant scroll bar.
- BUG ID 4991 - Adding a favorite directory without type using `menu.py` and the manual instructions doesn't work: favorite doesn't show up in any of the browsers.
- BUG ID 5001 - Hangs in Stabilize2D node.
- BUG ID 5012 - No error from version script up when Nuke script has not already been saved.
- BUG ID 5018 - Progress bars dialog follows the mouse.
- BUG ID 5027 - OFX not returning bounds correctly in some circumstances.
- BUG ID 5035 - Add Python binding for Shuffle knob.
- BUG ID 5039 - A script with Color > Math > Expression node hangs Nuke.
- BUG ID 5041 - User knobs are added at the bottom of the list, rather than after the currently highlighted knob.
- BUG ID 5047 - Assigning hotkeys to the Node Graph right-click menu doesn't work, whilst those assigned in the Application menu do.
- BUG ID 5053 - Undo not working properly in the Curve Editor if multiple curves are selected.
- BUG ID 5054 - TransformGeo: 'uniform scale' control affects rate at which mouse drags an object.
- BUG ID 5064 - Write node: .mov codec selected changed when script reloaded.
- BUG ID 5088 - Tracker: speed is considerably slower in Nuke 5 than in Nuke 4.
- BUG ID 5089 - `Nuke.tcl` should return the string not a boolean.
- BUG ID 5090 - Make frame in the Curve Editor respect the timeline setting for global/input/custom when framing expression driven curves.
- BUG ID 5111 - Merge: 'mask' knob reference in existing scripts not auto-fixed as 'mask-Channel'.
- BUG ID 5112 - Converting a gizmo into a Group and saving it as a Nuke script or gizmo sets knob values to 0.
- BUG ID 5115 - Colour picker not able to select from current buffer if placed below an 'over' node with two image inputs.

- BUG ID 5123 - Framecycle this script doesn't replace %V with the view name, so Framecycling a source image node directly can fail in stereo mode.
- BUG ID 5133 - Expression field in Curve Editor jumps to the end of expression when entering characters.
- BUG ID 5145 - Changing ScanlineRender's 'filter' control to 'Impulse' crashes Nuke.
- BUG ID 5167 - Copy node: the 'from->to' label for multiple channels no longer displayed on the node.
- BUG ID 5168 - Perspective texture not displayed correctly in OpenGL.
- BUG ID 5173 - Menu items added to the Nuke Application menu should not get added in Node Graph right-click menu.
- BUG ID 5175 - Bezier not showing animation dot on the node in the Node Graph.
- BUG ID 5184 - Make the JoinViews node work on any kind of input, not just image inputs.
- BUG ID 5190 - Avi files not recognised as QuickTime on Mac.
- BUG ID 5198 - Python panels crash if Cancel is hit and if a value is queried.
- BUG ID 5210 - Write node will not render out .exr / .sxr file formats with stereo output ticked, causing a crash.
- BUG ID 5211 - AutoCrop on Write node does not support stereo views.
- BUG ID 5217 - Python node creation: old control panels reappear and float briefly before new panels go to the Properties Bin.
- BUG ID 5238 - OFX plug-ins: not calling SyncPrivateData before a save/copy.
- BUG ID 5245 - QuickTime fails from command line passing argv 1 to Write node.
- BUG ID 5246 - QuickTime: random missing images when passing argv 0 to Read node.
- BUG ID 5249 - Geometry > Cube: the 'selectable' control disabled still allows mouse picking/moving.
- BUG ID 5293 - Wrong Python object type gets returned for menu item (returns a "Menu" object instead of the appropriate "Menu", "MenuItem", or "MenuType").
- BUG ID 5351 - Layout > Toggle Hide Floating Viewers does not work if Viewer is in full-screen mode.
- BUG ID 5355 - Proxy resolution mismatch : Bezier and other Draw nodes scaling incorrectly.
- BUG ID 5373 - Extracting a node that has multiple outputs does not disconnect from all the outputs.
- BUG ID 5392 - Cloned Read node does not correctly set the on/off state of the postage stamp.
- BUG ID 5394 - Certain TCL panel code not drawing properly in Nuke 5.
- BUG ID 5403 - User-coded (TCL, Python) file browsers do not open by default in specified directory.
- BUG ID 5406 - Postage stamp redraw is extremely slow.
- BUG ID 5420 - 3D > Modify > UVProject: spherical and cylindrical projection appear incorrect.
- BUG ID 5424 - Rotating a Bezier shape: the transform handle doesn't move but the points spin out of control.

- BUG ID 5429 - Add compiler version information for each platform to the makefile included in Documentation\NDK\examples.
- BUG ID 5445 - Permissions: if menu.py in NUKE_PATH is not accessible, you get seg fault on start-up.
- BUG ID 5487 - Creating a Dot node via menu or hot key should default to NOT open its control panel.
- BUG ID 5520 - Reword 'help' for 'nuke.createNode' to clarify 'inpanel'.
- BUG ID 5646 - Curve Editor's right-click Loop function (Predefined > Loop) only creates loops above frame 1.
- BUG ID 5749 - Python: nuke.channels() returns a list of <NULL> items.
- BUG ID 5891 - Write node: QuickTime advanced codec settings don't get saved for Avid DnxHD codec.
- BUG ID 5900 - Write and Read nodes: DPX edge_code - code put in our Write not then appearing in our Read of that file.
- BUG ID 6021 - Animation menu: Copy Values, Copy Animation, and Copy Links not working on multi-part knobs.
- BUG ID 6025 - Simple script crashing when viewing an IDistort node.
- BUG ID 6036 - Extra floating panels appear when loading layouts.
- BUG ID 6040 - Ocula: Write node set to 'all' channels does not write out the correlated view correctly.
- BUG ID 6042 - TCL documentation missing on Windows.
- BUG ID 6046 - Left/right view mapping for paint strokes lost on script re-load.

See also *Bug Fixes Logged During the Beta Period* on page 15.

Known Bugs and Workarounds

- On Mac OS X 10.5 (Leopard), when the Viewer is set to the Crystal Eyes 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 Crystal Eyes 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.
- BUG ID 3594 - Sampling colours for Primatte causes a double free or corruption crash. This is caused by incompatible libstdc++ library on the Nuke running environment in question. It's a problem with some Linux distributions, notably Ubuntu 6.06 and CentOS 4.4, and the libraries shipped with them. As a workaround, we ship the libraries we build Nuke against, so you can force your operating system to use these. To do so, set the LD_LIBRARY_PATH environment variable to point at the libraries.
- 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 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 5898 - Text on nodes on the Node Graph is fuzzy, hard to read, and loses lines occasionally.
As a workaround for this, font smoothing is turned off for Nuke during install on Mac OS X. You can also do this command manually from the terminal:
defaults write uk.co.thefoundry.Nuke5.1v1 AppleFontSmoothing -int 0
To re-enable font smoothing, you can use:
defaults write uk.co.thefoundry.Nuke5.1v1 AppleFontSmoothing -int 1
- 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 5966 - PLE: gizmos exported out of PLE are not read back in through Other > All Plugins > Update.
- BUG ID 5998 - In full-screen mode, the colour picker window goes behind the main window when using Linux Gnome window manager.
- BUG ID 6115 - The exr reader needs the Nuke cache directory to exist, otherwise it will fail with an error message of "exr reader failed: no such file or directory". As a workaround, create your Nuke cache directory.

Bug Fixes Logged During the Beta Period

5.1v1b7 Bug Fixes

- Fixed gamma shifts and worked around byte order problems for the BlackMagic codec on Intel Macs.
- Autolabel for mask from input now displays correctly.
- BUG ID 5103/5328 - C++ NDK is missing from distribution.
- BUG ID 5745 - Mask setting lost on load of script.
- BUG ID 5892 - Error message when adding Truelight v3 node; "Unrecognized cube generation option".
- BUG ID 5905 - 'Node name background' transparency options in the Node Graph preferences are not working on non-square nodes.
- BUG ID 5926 - OFX message suite not returning values correctly: 'OK' returned instead of 'Yes'/'No'.
- BUG ID 5929 - Changes to channels in gizmos not being saved correctly.
- BUG ID 5947 - Modifying user knobs messes up the Revert/Cancel/Close buttons that appear on floating properties panels.
- BUG ID 5957 - Save new version Python script failing with 'takes 4 arguments (3 given)' error message.
- BUG ID 5962 - Adding a user knob to an Axis node crashes Nuke.
- BUG ID 5965 - Read node: can't version up and down on a proxy file name when no full resolution file name exists.
- BUG ID 5968 - Some OFX plug-ins rely on being told to reprocess if the input is changed. If an upstream node is disabled rather than removed, the message doesn't seem to be sent.
- BUG ID 5978 - ScanlineRender: UV modes seem to be broken.
- BUG ID 5984 - Custom plug-ins crashing when calling `Op::error_message()` due to a type mismatch.
- BUG ID 5999 - Color > Exposure node values not ganging correctly when the slider is dragged.

5.1v1b6 Bug Fixes

- Fix for Node Graph arrows changing colour when zoomed out.
- Fixed a crash that occurred when setting a transform knob to a single number using TCL.
- DPX and QT writers now accurately write pixel aspect ratios that are a multiple of 1/3.
- BUG ID 5377 - Bezier shape offsetting from mouse pointer during transform.
- BUG ID 5467 - Tracker connected after Ocula's DisparityGenerator plug-in causes Nuke to crash upon running the track.
- BUG ID 5541 - Pressing the ' key with Viewer in the first Node Graph pane results in the entire Nuke interface being hidden.
- BUG ID 5566 - No way to remove curves from Curve Editor if you delete the node to which they relate.

- BUG ID 5593 - Other > StickyNote node does not get added to the Node Graph at the current mouse position but gets offset.
- BUG ID 5610 - Render > Flipbook progress doesn't update unless you drag the progress bar around.
- BUG ID 5613 - Versioning Read nodes with Alt+up/down arrow was not working properly. You can now version Read nodes in the following ways, for example:
/v01/test_v02.jpg > /v01/test_v03.jpg
/v01/test_v01.jpg > /v02/test_v02.jpg
/v01/test_v0001.jpg > /v02/test_v0002.jpg
- BUG ID 5618 - Connecting a Read node to a Viewer node causes Nuke to crash intermittently.
- BUG ID 5679 - Not shipping the 64-bit Linux libraries.
- BUG ID 5683 - When making the Viewer fullscreen and back, you get an error in terminal: "QOpenGLPaintEngine: Failed to create fragment programs".
- BUG ID 5684 - The NDK Documentation link for C++ include files is broken.
- BUG ID 5728 - Bezier handle angles wrong for anamorphic images.
- BUG ID 5702 - Playback is slow on Windows when the Node Graph or Curve Editor is open.
- BUG ID 5730 - Viewer's playback speed does not match the frame rate set.
- BUG ID 5733 - Render cancels after a few frames with certain scripts.
- BUG ID 5735 - Numerous error messages when loading a certain script.
- BUG ID 5754 - A script with a LightWrap node rendering incorrectly.
- BUG ID 5769 - Unable to open any saved scripts in Nuke terminal mode; returns syntax error."
- BUG ID 5771 - Gizmo knobs lose value when reopening a saved script if the knob value has a certain default.
- BUG ID 5776 - Gizmo user knobs load in incorrectly compared to Nuke 5.0.
- BUG ID 5779 - Blur: corruption in the PLE (Personal Learning Edition) mode.
- BUG ID 5780 - Many nodes not working on Mac OS X (PPC) in PLE mode, producing the following error: "no ColorCorrect node Class".
- BUG ID 5788 - In certain circumstances, Viewers are not reusing Viewer1 panel when it's empty.
- BUG ID 5801 - No error messages from loading script that should error.
- BUG ID 5808 - Problems using Wacom tablets in Nuke.
- BUG ID 5820 - Write: rendering an xpm file causes a crash.
- BUG ID 5881 - On Mac OS X, the main window title bar is hidden behind the menu bar.
- BUG ID 5887 - Colour picker disappears behind the Nuke window when focus is lost.
- BUG ID 5893 - Node rename dialog doesn't disappear until the Node Graph is hidden or restored.
- BUG ID 5895 - Right-clicking and selecting "Revert knobs" causes mask to be turned on and set to alpha.
- BUG ID 5897 - On Linux, floating windows and some dialogs go behind fullscreen windows.

- BUG ID 5903 – Python methods use 32-bit floats as opposed to 64-bit, leading to loss of precisions.
- BUG ID 5916 – NDK: OFX crash when calling messageSuite from createInstance (in Examples/Basic/basic.cpp).
- BUG ID 5918 – Views > Stereo > ReConverge node should alter disparity fields according to the changes it produces.

5.1v1b5 Bug Fixes

Internal release only.

5.1v1b4 Bug Fixes

- BUG ID 5034 – Installer doesn't add an icon to the Desktop when the option to do so is selected.
- BUG ID 5147 – Text node: several font path problems.
- BUG ID 5218 – Creating a node via Python in a separate thread crashes Nuke.
- BUG ID 5462 – Pivot point adjustments cause exaggerated and incorrect transforms.
- BUG ID 5515 – Card node: bicubic deform produces strange results after reverting the changes made to the node.
- BUG ID 5609 – 'Set knobs to default' doesn't obey the knob defaults set.
- BUG ID 5611 – Geometry disappears on some frames when viewing a test script.
- BUG ID 5649 – FrameHold: incorrect frame displayed in control panel, leading to errors in expression linking.
- BUG ID 5666 – Lines drawn across nodes in the Node Graph.
- BUG ID 5673 – Python: `nuke.Root()` creates a Group node.
- BUG ID 5683 – When making the Viewer fullscreen and then back again, you get an error in terminal: `QOpenGLPaintEngine: Failed to create fragment programs.`
- BUG ID 5688 – Card node: undo doesn't revert back to previous state but takes smaller steps in between.
- BUG ID 5702 – Playback is slow on Windows when the Node Graph or Curve Editor is open.
- BUG ID 5703 – Read nodes not applying 'default' colorspace correctly.
- BUG ID 5725 – Python: crash when using `scriptclear` and animating a subsequent value.
- BUG ID 5733 – Render sometimes cancels after a few frames with some scripts.
- BUG ID 5746 – Fbx frame rate not used from the fbx file.

5.1v1b3 Bug Fixes

- 3D: Fixed a rare texture map striping problem.
- Fixed missing elbows in the Node Graph when Ctrl/Cmd key held down.
- BUG ID 4978 – Toolbar, menu, and hot key items cannot be replaced by a second call using the same name.
- BUG ID 5032 – Draw > Text node error; font extension ttf needed in path for Mac OS X 10.5 (Leopard).

- BUG ID 5101 - Right-click "Set knobs to default" does not reset knobs which rely on the size of the input image correctly.
- BUG ID 5157 - FBX readers do not error out when incorrect path is specified in the file name.
- BUG ID 5196 - Nuke crashed when cancelling a terminal mode render with Ctrl+C. On Windows, Ctrl+C exits immediately. Ctrl+Break cancels a render without exiting if a render is active, or exits if not. On OS X and Linux, Ctrl/Cmd+C always exits.
- BUG ID 5209 - Bicubic Card node: adding a subdivision should not change the existing shape of the image unnecessarily.
- BUG ID 5235 - Write node: duplicate QuickTime controls between Write control panel and advanced codec settings.
- BUG ID 5263 - On Mac, closing floating HueCorrect controls causes a crash.
- BUG ID 5268 - File handles left open when running a test script.
- BUG ID 5296 - Toolbars in custom menu.py are not restored in default layout.
- BUG ID 5312 - Views > Split and Join not working.
- BUG ID 5316 - When typing 'x' and a TCL command, Nuke displays the result twice.
- BUG ID 5326 - Undoing a Bezier point with certain steps causes the Bezier tangents to change (on Windows).
- BUG ID 5327 - Crop parameters are set to 0,0,0,0 on creation instead of using bounding box from incoming stream.
- BUG ID 5339 - 3D > Geometry > Card : bicubic subdivide jack can be moved off the object's surface.
- BUG ID 5344 - Project settings > Views tab > "Use colours in UI" not working; Node Graph arrows for different views are not set to the appropriate colours.
- BUG ID 5348 - Curve Editor: Predefined > Negate produces an error.
- BUG ID 5356 - Copy and paste crashes Nuke in some cases with stereo enabled in the Project settings.
- BUG ID 5410 - Keys set on bicubic Card node are not indicated in the Viewer timeline.
- BUG ID 5415 - Color > Truelight: clicking 'show commands' button causes an error message.
- BUG ID 5433 - Full frame processing runs over and over when an OFX node feeds to an Anaglyph node.
- BUG ID 5446 - Read: rgba image reader not functioning.
- BUG ID 5450 - Stereo exr Write puts single view channels into file for each view.
- BUG ID 5491 - Python: nuke.menu("Nuke").findItem() fails when menu name has an & accelerator in it.
- BUG ID 5500 - When using addFavoriteDir in your menu.py, you get a "folders.nk not found" error message.
- BUG ID 5512 - Windows seems to limit to 2 gigs of RAM regardless of what is available.
- BUG ID 5513 - Draw nodes: leaving a 1 pixel gap around the side of all drawn images.

- BUG ID 5516 - Pressing a number key to hook up a Viewer without any nodes selected results in two error messages: "no node selected" followed by "error whilst executing script".
- BUG ID 5528 - Creating a Node Graph 'elbow' dot should not open the dot's control panel.
- BUG ID 5540 - Python: showname.py nukescrpt errors out if it's run when there is currently no node selected.
- BUG ID 5545 - Crashing with Tinder OFX plug-ins when dragging the Viewer timeline.
- BUG ID 5552 - Filter > Convolve node not working.
- BUG ID 5560 - File Browser: it was not possible to preview frames higher than frame 100.
- BUG ID 5576 - Gizmos losing knob settings and going back to defaults on a script load.
- BUG ID 5591 - Python: crash trying to setValues on an enumeration knob.
- BUG ID 5607 - Write node not updating file type when in proxy mode.
- BUG ID 5616 - No errors reported from incorrect scripts run from menu entries.
- BUG ID 5617 - Cannot pick tabs or grouped knobs for user knobs anymore.
- BUG ID 5622 - Output from one Script Editor panel pops up in all open Script Editor panels.
- BUG ID 5625 - Transform > BlackOutside produces two rows of black at top edge.
- BUG ID 5651 - Adding a Python expression such as [python -eval sfjsdhfk0] crashes Nuke.

5.1v1b2 Bug Fixes

Internal release only.

5.1v1b1 Bug Fixes

- BUG ID 3632 - Drag-adding a ColourTransfer between a Colourbars and a Viewer crashes Nuke.
- BUG ID 3642 - ColorTransfer: crashes when just input B is connected.
- BUG ID 3658 - ColorTransfer did not handle inputs with no channels, crashing a test script.
- BUG ID 4920 - Colour picker disappears when clicking on it while viewing 3D.
- BUG ID 4921 - Deleting a node after a colour picker has been closed causes a crash.
- BUG ID 4926 - Bundle older version of OFlow so that scripts that include the older version can still be opened.
- BUG ID 4943 - When Nuke loses focus, progress bars (for example, Write progress) disappear.
- BUG ID 5022 - The Shift and Shift+Y hotkeys for connecting nodes are confusing.
- BUG ID 5024 - Flipping between views in the Viewer throws away memory cache.
- BUG ID 5025 - Remove bracket/parentheses parsing from autolabel script.
- BUG ID 5058 - Creating a Group node doesn't create input nodes for disconnected inputs.
- BUG ID 5059 - Closing the Viewer that opens with Nuke -v causes Nuke to crash.
- BUG ID 5060 - Running Nuke in Viewer only mode (-v) doesn't work in Windows XP.
- BUG ID 5065 - Heap corruption error when deleting a Viewer when the mouse is over it.
- BUG ID 5066 - TCL called from Curve Editor menu being executed as Python.



- BUG ID 5067 - Toggle.py script is failing when trying to disable a Backdrop node.
- BUG ID 5068 - UVs getting dropped by TransformGeo-based nodes.
- BUG ID 5070 - Cascading dialog boxes flicker when opening image sequences or MergeExpression node in Windows XP.
- BUG ID 5080 - Tabs in gizmos misnamed.
- BUG ID 5091 - Running Nuke in Viewer only mode and quitting brings up a notification about saving changes.
- BUG ID 5092 - Running Nuke in Viewer only mode should switch the Viewer to 'input' time-frame and pick up the input range.
- BUG ID 5094 - Python binding missing for MultiArray_Knob.
- BUG ID 5099 - Spitting unnecessary messages to terminal when rendering OFX plug-ins. This now only happens in the verbose mode, which can be activated by using the command line option -V.
- BUG ID 5100 - Render and write of a test script is different in different OS's.
- BUG ID 5109 - User knob disappears when having chosen not to delete it.
- BUG ID 5119 - Selecting File > Open and Cancel brings up a Python exception.
- BUG ID 5151 - Deleting the Viewer from a test script while the Viewer is rendering causes a crash.
- BUG ID 5153 - Nuke.dependencies() causes a crash.
- BUG ID 5159 - Ctrl+C or Cmd+C doesn't copy text from the Script Editor's output pane.
- BUG ID 5164 - OFX plug-ins are being given the wrong view in some circumstances.
- BUG ID 5172 - Start-up license error message is slightly obscured by the Nuke splash screen.
- BUG ID 5183 - Add 'Input On\Off' and 'Postage Stamp On\Off' menu entries in the Edit > Node menu.
- BUG ID 5243 - The Node Graph slides/scrolls in opposite direction to mouse movement or after pen up using wacom.
- BUG ID 5262 - Deletion/hiding of user knobs from Python.
- BUG ID 5281 - Creating nodes via nuke.nodes won't take all knobs.

DEVELOPER NOTES

Here are the changes relevant to developers.

Changes between Nuke 5.0v2 and Nuke 5.1v1

For writers of DDImage binary plug-ins:

The "Animation" object has been removed. This was done so that the implementation can be improved. All the useful actions can instead be done by using calls on Knob objects.

New LookupCurves object and LookupCurves_knob() that allows you to write operators that use user-edited look-up curves. This replaces Animation_knob and requires operators using that to be rewritten.

Several include files have been renamed to get around bugs in compiler and build systems where all files must have different names:

DDImage/math.h has been renamed DDImage/DDMath.h

DDImage/string.h has been renamed DDImage/DDString.h

New predefined channel numbers that are output by Ocula:

Chan_Stereo_Disp_Left_X

Chan_Stereo_Disp_Left_Y

Chan_Stereo_Disp_Right_X

Chan_Stereo_Disp_Right_Y

New predefined channel number output by the "inject mask" option on a number of operators:

Chan_Mask

Note that this is called mask.a in Nuke on the assumption that coloured masking will be added in the future.

New very useful functions to go between channels and layers:

```
const char* getName(channel);
```

```
const char* getLayerName(channel);
```

```
unsigned colourIndex(Channel); // return 0=red, 1=green, 2=blue, 3=other
```

```
Channel brother(Channel, index); // return channel from layer with given colourIndex()
```

```
ChannelSet::addBothers(Channel, n); // add all the brothers from 0..n-1 to this set
```

The ColorLookup Op base class has been changed to manage only four curves and apply them to any number of channels based on the colourIndex() of these channels. This will require any subclasses of them to be rewritten.

`Execute::generateOp()` takes an optional view number as well as a frame number.

`Hash::append(const std::string&)` has been added so you can hash stl strings without using `c_str()`.

`Knob::name()` and `Knob::label()` are `std::string` now. Use `name_c_str()` and `label_c_str()` to get the previous values. Please do not use `strcmp` for the knob names, but use `knob->is("name")`. This is portable to Nuke 4.8 as well.

`Obsolete_knob(cb, new_knob_name, old_knob_name)` allows you to alias an old knob name to a new one, so both code that sets and reads the old knob continues to work. Note that the arguments are backwards to what you might expect. This was done to match some other knob calls.

The "node" argument to `NukeWrapper` constructor is removed. Just remove it from your code. It is not legal for it to be different than the `lop`'s node so it initializes it from the `lop` now.

The "flags" to the `NukeWrapper` have been removed as they were quite confusing. Instead, you can now preset some controls by adding some methods after the constructor:

```
new NukeWrapper(op)->noMix() // turn off mix control
->noChannels() // turn off channelSet control
->noMask() // turn off masking and the mask input
->noUnpremult() // turn off unpremult control
->channels(Mask_xyz) // initialize channelSet to something other than Mask_All
```

`Op::detach()` and `Op::attach()` are called when the user deletes or undeletes a node. This can be used to free resources. Note that this is a temporary measure. Future versions of Nuke will actually call your destructor, so make sure your destructor works, too.

New `ReadWriteLock` object in `Thread.h`. This can be locked by either exactly one writer, or by any number of readers.

You may have directly addressed the "DD::Image::Animation" object in your plug-ins. This is no longer allowed, as it has been hidden in order to allow the implementation to be altered without breaking plug-ins. Some of the proposed alterations are to allow compiled expressions, to allow stereo view numbers, and to allow a single curve + expression to produce all the fields of a knob using Python vectors. If you used the `Animation` object to set keyframes on regular knobs, you should instead use the methods on the knob itself that sets and/or examines keyframes:

```
Knob::set_animated()
```

```

Knob::is_animated()
Knob::clear_animated()
Knob::set_value_at(value, time) // sets keys if animated is true
Knob::set_values_at(values[], time) // for multi-channel knob
Knob::remove_key_at(time)
Knob::is_key_at(time)
Knob::set_expression("expression")
Knob::getNumKeys()
Knob::getKeyIndex(time);
Knob::getKeyTime(int index)
Knob::getDerivative(time, int channel=0)
Knob::getNthDerivative(n, time, int channel=0)
Knob::getIntegral(time1, time2, int channel=0)

```

Much more common was use of the Animation object to add what was called an "Animation_knob" to the control panel, which presented an curve editor and was usually used to edit a lookup curve. This was renamed to "LookupCurves_knob" to better reflect how it is actually used, and the api was changed to avoid any assumptions about the number and type of objects that are stored:

Old:

```

static const char* curvedefaults[] = {
    "name1", "y C 0 1",
    "name2", "y C 0 1",
    0
};

```

New:

```

static const CurveDescription curvedefaults[] = {
    {"name1", "y C 0 1"},
    {"name2", "y C 0 1"},
    {0}
};

```

Old:

```

MyClass {
    const Animation* curves[2];
    MyClass() {curves[0] = curves[1] = 0;}
};

```

New:

```

MyClass {
    LookupCurves curves;
    MyClass() : curves(curvedefaults) {...}
};

```

Old:

```
double v = evaluate(curves[n], x);
```

New:

```
double v = curves.getValue(n, x);
```

Old:

```
MyClass::knobs(Knob_Callback cb) {  
    Animation_knob(cb, curves, curvedefaults, "name");  
}
```

New:

```
MyClass::knobs(Knob_Callback cb) {  
    LookupCurves_knob(cb, &curves, "name");  
}
```

Writer::status(x) has been renamed **Writer::progressFraction(x)**. Also, you can call **Writer::progressFraction(a,n)** where a is the current line you are working on and n is the total.

Writer::knob_changed() virtual function added, so file Writer objects can respond to the user moving any knobs on the Write node.

