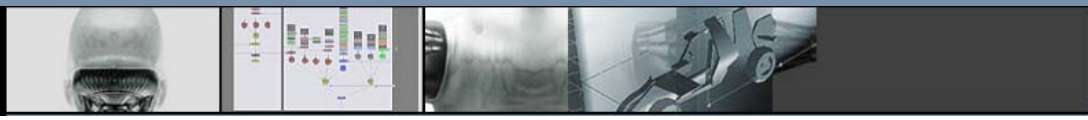

Release Notes, 4.8v1

This is a maintenance release of Nuke with some of the features you will see in the forthcoming Nuke 5.0.

Version

Nuke 4.8v1

Release Date

05 February 2008

Requirements

Supported operating systems:

- Mac OS X (10.4).
- Windows XP SP2.
- Linux CentOS 4.5.

New Features

- Added support for 1023 channels instead of 64. Like before, channels that are not used have no effect on speed.
- Any file named *template.nk* in the plugin path is loaded instead of an empty script when you launch Nuke or select **File** > **New** or **File** > **Close**. This allows you to save LUT and other setups and favourite arrangements of nodes.
- You can create an unlimited number of new global lookup tables (LUTs). These LUTs are available in the **colorspace** pulldown menu of Read and Write nodes' and Viewers' controls. You can also choose any LUT to use as default for 8-bit, 16-bit, log, and float files. For more information, see the *Configuring Nuke* section in the user guide.

- Added interactive lighting tools in the 3D compositing workspace, with support for four types of lights: direct, point, spotlight, and environment light. Direct, point, and spotlight are standard 3D lighting objects. The environment lights generate image-based lighting via HDRI to influence various surface properties. Lighting may be masked according to channel data or rendered images. 3D display was also improved with high-quality anti-aliasing.

- Added two new 3D > Shader nodes: Phong and BasicMaterial. Phong uses the Phong algorithm to provide accurate shading and highlights. BasicMaterial is a combined node for diffuse, specular, and emission materials.

- Added new syntax to the expressions that you can put into knobs:

  ### Field names optional

  The expression on field N of a knob will refer to field N of another knob unless a field is specified.

  Before, if you needed one xy knob to refer to another, you had to set the x expression to "otherknob.x" and the y to "otherknob.y". Now you can enter just the string "otherknob" in both expressions. This is convenient as the string is now identical for all fields and you can copy & paste it.

  This also works for the derivative and other functions described below.

  ### Derivatives

  Rather than having to type "(knob(x+.1)-knob(x))/.1", Nuke directly computes derivatives for you. These are exact if your curves have only keyframes. If you use a math expression it will do numerical deriviatives, with pretty good accuracy up to the third derivative.

  knob.field.derivative = derivative at the current frame

  knob.field.derivative(x) = derivative at frame x

  knob.field.derivative(n,x) = Nth derivative at frame x

  knob.field.derivative(0,x) = same as knob.field(x)

  ### Integrate

  knob.field.integrate(x1,x2) = numerical integral between x1 and x2

  This is done by sampling. Future versions may do keyframed curves with much better accuracy.

  ### Inverse

knob.field.inverse(y) = value of x such that knob(x) == y.

knob.field.inverse = value of x such that knob(x) == current frame.

This is for inverting color lookup tables. Knob must have a zero or positive derivative everywhere, and this works best if the answer is between 0 and 1.

Improvements

These are the improvements made to existing features since the previous release.

- The monitor LUT in the root settings is now scaled from 0-1 rather than 0-255, so that import/export and copy/paste of LUTs is easily compatible with other Lookup functions in Nuke. No backward compatability at all. The previous monitor LUTs are ignored except for a setting to linear.
- Increased size of 3D render high antialiasing kernel to overlap surrounding pixels, greatly increasing the appearance of specular edges. Also addressed comments regarding how similar low and high antialiasing appeared.
- Improved playback speed on OS X.
- Axis_Knob draws axis at pivot location, not origin. This mimicks Maya/3DMax/XSI/Modo/etc. To make adjusting the pivot easier, CTRL/META allows GUI adjustment of pivot.
- Improved the Curve editor:

  Drag movement is constrained to horizontal and vertical, unless you hold down **Ctrl** (**Command** on OS X). You have to move several pixels before the direction to constrain is selected, so you are more likely to get the direction you expect.

  You can no longer select slope handles that are invisible. Selecting slope handles does not deselect keys.

  You can now select and move points that are not on the yellow selected curve.

  Typing "**=**" pops up the expression entry dialogue.

  Generate and **Import Ascii** actions remove expressions. This makes it easier to replace slow, complex expressions in the LUTs with key frames.
- Created additional NDK examples and updated plugin compilation instructions.
- Improved tracker accuracy on noisy images.

- SplineWarp directly samples input image rather than renders a polygonal model, resulting in much smoother warps. Also, SplineWarp control panel now has a **show** menu item that lets you choose which image and curves are displayed.
- Drawing ops now offer a choice of what to clip to: none, bbox, format, or both.
- Added new merge operation called "hypot" to Merge2. This is the same as the "diagonal" merge in the older Merge op. It does hypot(A,B) which is sqrt(A*A+B*B).
- Added a command line switch -s to set the thread stack size.
- Truelight now only reloads if the knob values change.
- Truelight: Uses $TRUELIGHT_PROFILE as the default profile to load, to match the $TRUELIGHT_DISPLAY environment variable.
- Added an additional default LUT field for Viewer, so that it can be set separately from other monitor-drawn images. For example, if you are using a Viewer Input Process that converts all the way to monitor space you will want to set the Viewer LUT to Linear (for no change) but still have the Monitor LUT set to sRGB so that postage stamps, textures in GL, etc. look reasonable.
- Added help line for stack size.
- Curve editor menu item to open up curve editor without setting a key even if unanimated.
- Minor improvement to thread locking on Windows.
- Made the "sample node channel x y" TCL command produce a point sample. To get the previous filtered result, use "sample node channel x y 1 1".
- Render speed improvements.
- Color/Math/Expression can now get the value of pixels at arbitrary locations in the input image:
  - red(x,y) will return the value of the red channel at a point sample at x,y. Note that integers are the corners of pixels, you may need to add .5 to get the pixel you expect.
  - red(x,y,dx,dy) will return a cubic filtered sample centered at x,y and dx,dy wide and tall. Sizes less than 1, including zero, produce the same filtered result as 1.
  - "red" can be replaced with the name of any input channel, including a layer.channel name. The merge expression can prefix "A." or "B." to indicate what input to use.

- Added the A bounding box option to Merge.
- Performance fixes improvements especially in STMap and random other operators (bug 2279, bug 3342).

Bug Fixes

- No longer prints the "license file not found" help text if the startup error is something other than license not found. Fixed crash when unable to write license log file.
- Removed the defunct FloodFill node from toolbars.
- Fixed a typo that caused the loading of "standard" preferences to produce an error message.
- No longer leaves 256 file descriptors open when first creating cache (Windows).
- Color/Math/Expression crashed if an expression had more than about a hundred terms (each number and operator is a term). The number of terms is now unlimited.
- Fixed Axis Knob to draw its motion path again.
- Labels in Shuffle are no longer blinking.
- Expression node no longer crashes on a new expression entry due to the output channel not being requested from the input.
- 3D view can now be locked to an orthographic camera. Before it would force a perspective view.
- The numvalue command was not returning the same frame number as the value command when there was a time offset.
- Fixed a multi-threading related crash that could occur while running the Tracker.
- Fixed cursor not getting displayed in Paint glitch.
- Fixed light aliasing by always doing lighting calculations, even when the surface angle is away from the camera.
- Made the keypad arrow keys work as "nudge" keys in the curve editor again.
- Increased the high anti-alias span/pixel hit test threshold to 1.25 from 1.0
- Truelight: Scripts saved with the "advanced" checkmark turned off will still have it turned off when reloaded.

- Minor fix for truelight so that turning on advanced is faster.GridWarp crashed on startup in some circumstances.
- VectorBlur with forward algorithm produced the wrong image if the viewer was zoomed in or panned or the ROI was on (part fix for bug 2351).
- Fix crash on startup in Leopard.
- Enforceed a minimum pthread stacks size of 8Mb as OSX has a default of 512k and the extra number of channels was causing it to run out of stack.
- Errors in the preferences file caused it to abort. Changed to instead parse to the end of the file.
- Fix so Card3D zooms to the correct format in the viewer.
- Color/Expression would produce wrong results or crash if not all the channels were requested from it.
- Fixed version number for OFX plugin cache.
- Truelight node did not write the display_file out if it happened to be the same as the user's TRUELIGHT_DISPLAY. This caused the saved script to not work as expected on a render farm.
- File knobs on controls default to normal files, with no image preview or sequence collapsing.
- Fixed crash if ColorLookup had expressions in the curves.
- 3D objects: Empty bounding box was getting transformed and appended when it should have been ignored.
- 3D: Card crashed if input was disconnected.
- 3D: OpenGL preview broken when using merged materials.
- Fixed a cause of random tracker crashing.
- Nodes with more than 10 or so disconnected arrows drew the ones that should be on the right edge in the wrong place.
- Tracker "Grab when error" fixed to grab the new search feature correctly.
- Removed Fog node from menus as it appears to be non-functional.
- 3D: Geometry caches use hash values to find matching caches in other scenes - allows Motionblur to work despite the disappearance of objects on sub-frame scenes.
- 3D: Scene object remembers its frame number.

- Fixed a channel request bug in ScanlineRender node.
- Fixed a rare but deadly null pointer crash in Viewer image management.
- Fixed a bug in built-in LUT Cineon colorspace conversion.
- Fixed crash occuring when node owning 3D objects drawn in Viewer was deleted.
- Fixed TIFF reader releasing cached lines that were still in use.
- Fixed crash due to writing beyond the end of the cache line.
- Fixed a case of 3D objects in Viewer not updating when frame changed.
- Default stack size is now 16MB to accommodate needs of increased number of channels.
- We now configure open file limit to 1024 to prevent "too many open files" errors.
- Doxygen docs for sample() were running together with the next one, modified().
- GeoOp::add_object() didn't exit properly if the referenced object already existed.

  Allows SourceGeo types to modify their geometry without having to dump their cache completely.
- Added GeoOp::clear_rebuild() method to turn off rebuild bits as well as turn them on.

  This gives plugins better control over what geometry groups get rebuilt.
- Project3D must not restrict the UV range when the renderer is in UV mode.
- Channel selectors in Primatte were using the wrong input to decide what channels were available.
- If ~/.nuke/init.tcl contained a "plugin_addpath" command, saved preferences and other files would be added to that directory rather than ~/.nuke.
- 3D: Corrected a bug with the Phong OpenGL diffuse material.
- 3D: Made it calculate the correct request area if more than one Project3D node is used on a piece of geometry.
- 3D: UV bound fix.

- Date did not appear in the File Browser for the first directory in a list with directory names that looked like a sequence of files (i.e. with a number in them and more than 2).
- Code put in to stop Escape from closing the viewer if it is typed to abort an execute. Also stopped it from closing if the viewer was currently updating.
- Fixed mov file readers so that they are not destroyed and reopened on each frame. This should improve reading Quicktime files a lot.
- Documentation: Updated instructions for Oflow.
- 3D: Changes to node color chips no longer tint OpenGL textures.
- Truelight: In earlier 4.8 beta builds, the option to convert back to linear was using the current viewer LUT. We now always use sRGB (node rendering should never be affected by viewer settings).
- Fixed a general cause of "This is probably a coding error in an Op" errors.
- Crash when changing visible layer in Viewer while updating.
- 3D: Fixed bug where point-cloud mode at the primitives frame, the reader would read the full geometry file twice, creating duplicate vertex attributes.
- ObjWriter fixes for failing to update the object when executing over a range of frames.
- Search for OFX plugins in /usr/OFX/Nuke.
- BUG ID 1699 - Curve Editor: You can now move a set of selected points on the curve if they all have the same Y value.
- BUG ID 1819 - Memory allocation failure not caught for OFX plugins.
- BUG ID 1848 - Panning and 2D/3D mode flipping should not reset 2D zoom to autoscale.
- BUG ID 1980 - Crash when rendering 3D motion vectors if input textures vary on non-integer frames.
- BUG ID 2068 - "Start in" field of Desktop shortcut left blank, doesn't work for roaming profiles.
- BUG ID 2087 - Crash on Bezier Mix=0 + FilterErode due to FilterErode trying to erode a non-existent channel.
- BUG ID 2148 - Pressing Ctrl+A in the LUT editor caused an immediate crash.
- BUG ID 2150 - Curve Editor: You can now manipulate multiple points when all selected points have same value in one axis.

- BUG ID 2154 - "Bad format" error when reading a large tiff file.
- BUG ID 2173 - Rendering - Now shows most recent node reporting progress.
- BUG ID 2231 - Nuke crashes with exr's rendered out from XSI, Mental Ray.
- BUG ID 2279 - Performance: 3D playback/render is much slower on the Mac platform.
- BUG ID 2289 - Nuke crashes when importing a Photoshop file.
- BUG ID 2312 - ColorLookup doesn't apply imported ascii LUT as root LUT does. All LUTs are now 0-1; previous versions had monitor LUT in root settings in 0-255 range.
- BUG ID 2320 - EXR files only writing 16 channels.
- BUG ID 2343 - Selecting motion layer in this script crashes Nuke (attempt to access an invalid channel index).
- BUG ID 2356 - Built-in gizmos had name embedded and thus were always created with the same name, not incrementing number.
- BUG ID 2364 - QuickTime codec is stored as 4-letter name rather than a text name which differs between platforms. Note that old scripts will default to Motion JPEG and need to have codec reset since the fundamental problem being addressed makes it difficult/unreliable to know what the platform independent code is.
- BUG ID 2383 - Unhandled pixel format: BGRA on QuickTime files encoded with BlackMagic 10 bit codec.
- BUG ID 2430 - Random crash when working with gizmos or when using the enhanced flipbook.
- BUG ID 2436 - Drag & drop animation from unnamed user knob crashes Nuke.
- BUG ID 2442 - Add a second channel picker to Convolve to allow all channels to reference a single channel convolve shape.
- BUG ID 2447 - Crop node causing 3d system to be translated incorrectly.
- BUG ID 2460 - Bezier handles don't update properly with non-square pixel aspects.
- BUG ID 2469 - Expression nodes don't evaluate animation - only the value of the first keyframe is used.
- BUG ID 2496 - Nuke crashes when output of Convolve node is written to a file.

- BUG ID 2497 - TCL expression not updating.
- BUG ID 2501 - Write: Quicktime writes cause a crash if Quicktime is not installed.
- BUG ID 2510 - Killing a viewer (in 3D mode) now stops it calculating texture maps for 3D view in the background.
- BUG ID 2514 - Fog - not updating when parameters changed, and seems to be modifying depth channel in a bizarre fashion.
- BUG ID 2518 - The 'hypot' function doesn't work in the Expression node.
- BUG ID 2598 - LayerContactSheet (and ContactSheet) node repositions/ resizes contact sheet images unexpectedly.
- BUG ID 2639 - Paint node erroring in customer script with: "Interest(channels=f), request(channels=0). This is probably a coding error in an Op connected to this one's output".
- BUG ID 2673 - Error in Premult4 with specific script.
- BUG ID 2675 - Crashing in LayerContactSheet rendering.
- BUG ID 2677 - Text node expression is returning different values when same value is correct despite changing input connection.
- BUG ID 2682 - A "monitor" truelight display is now provided, so that the default settings of Truelight work without an error. (Just a copy of the "SonyHD" one.)
- BUG ID 2691 - ShuffleCopy doesn't allow you to see some layers and then crashes when viewing the test setup.
- BUG ID 2693 - Pausing all viewers will crash Nuke.
- BUG ID 2694 - OFlow and Kronos have no effect on transformed beziers or paint.
- BUG ID 2695 - Shuffle after OFlow give incorrect results.
- BUG ID 2698 - Nuke crashes while rendering with Tinder and specific script.
- BUG ID 2730 - Crash when show_settings command executed in terminal.
- BUG ID 2739 - File->TCL file... doesn't work.
- BUG ID 2746 - Beziers cannot copy & paste shape key frames.

- BUG ID 2747 - Nuke starting up with a root script path that has backslashes got them resolved as escape characters, screwing up the path (Windows only).
- BUG ID 2758 - Warper produces output that doesn't line up with controls.
- BUG ID 2759 - SplineWarp now calculates output bounding box accurately.
- BUG ID 2764 - Write: Crash when trying to write an output file (jpeg).
- BUG ID 2828 - Scanline renderer: Missing the previuosly present 'output velocity' knob.
- BUG ID 2845 - Filter > Emboss node only works at frame it was created on, all others just pass through unchanged.
- BUG ID 2856 - Alt - MMB zoom not working.
- BUG ID 2858 - Script crashing on an assert in ChannelSet.size().
- BUG ID 2859 - "Edit/Node/Paste knob values" should not paste node names, tile colours and such, just non-generic knobs.
- BUG ID 2861 - Crash with Shuffle node.
- BUG ID 2865 - Crash when using the first_frame expression in the first_frame knob of the framehold node.
- BUG ID 2878 - Dropping animation buttons on themselves is now ignored and does not produce a recursive expression.
- BUG ID 2920 - Nuke not releasing memory correctly on some scripts.
- BUG ID 2922 - Primatte crash with specific script.
- BUG ID 2923 - Renders fail on large formats.
- BUG ID 2930 - Less DAG information as to what is being called on when processing.
- BUG ID 2936 - Nuke 4.8 still seems to be altering/looking at the preferences4.7.nk file rather than creating a new 4.8 one.
- BUG ID 2953 - Problem in GeoOp Split/Merge (3D) - Nuke crashes when opening the stadium script.
- BUG ID 2972 - Bezier + Primatte : Beziers won't draw after adding Primatte, and Nuke crashes. (Fixed the crash. However, if there is a control in the control panel bin with an active color picker, it will still take precedence over other manipulations, like laying down Bezier points.)

- BUG ID 3001 - OFX cache always been rebuilt in non-interactive mode.
- BUG ID 3002 - Nuke - T_Caustic height map not working until you disconnect and reconnect it.
- BUG ID 3008 - GridWarp ignores background knob.
- BUG ID 3013 - Card3D, can't pick rotation (OSC).
- BUG ID 3022 - Tiff reader running out of memory with large format images.
- BUG ID 3057 - Nuke automatically changes file type in the Write node when file path is changed.
- BUG ID 3062 - Bad PNG file gives warnings in Linux.
- BUG ID 3068 - Bad PNG file crashes Nuke on Windows.
- BUG ID 3083 - Colorcorrect: Range tab no longer draws line based on sample colour underneath cursor when on viewer.
- BUG ID 3102 - 3D display colors: If the handle colour is left at default, solid+lines draws faces and edges in the same colour.
- BUG ID 3114 - Viewer doesn't update when wipe is active.
- BUG ID 3149 - Nodes not erroring out correctly.
- BUG ID 3171 - Customer script crashes Nuke on Windows with SplineWarp.
- BUG ID 3189 - Merge/Project3D: projection cropped when using Merge node.
- BUG ID 3190 - Transform: hitting '=' in a multi-field knob clears fields you don't input to.
- BUG ID 3202 - Rendering slower on Windows than 4.7.
- BUG ID 3220 - Normals - needs much more info on the tooltip as to how to use with various channels, etc.
- BUG ID 3225 - Normals: delete mode can result in seemingly uninitialised memory getting written out as an image.
- BUG ID 3245 - Pixel aspect not included in a knob hash resulting in incorrect cache hit.
- BUG ID 3260 - Fixed two array look up out of bounds errors in the animation point editing.
- BUG ID 3270 - Read: tif doesn't read into OS X or Linux (Windows ok).

- BUG ID 3295 - Exec in TCL scripts refusing to run since earlier 4.8 build.
- BUG ID 3302 - 3D nodes can append to 2D nodes through copy and paste.
- BUG ID 3304 - 3D modifier nodes seem to have started changing the geo draw colour to their own tile colour.
- BUG ID 3305 - SplineWarp crashing since earlier beta build.
- BUG ID 3311 - Channel request bug in Colorspace node.
- BUG ID 3312 - Invalid edit of node name updates references to it (with invalid name) before Nuke auto-corrects the edit.
- BUG ID 3316 - File handler is not closed when rendering is cancelled.
- BUG ID 3317 - Curve editor not drawing reference curves or expression curves correctly.
- BUG ID 3342 - Rendering slowdown in sampling operations since earlier 4.8 builds.
- BUG ID 3343 - Rendering slowdown between 4.7v4 and 4.7v5.
- BUG ID 3362 - Group: unnamed input crashes Nuke.
- BUG ID 3396 - Opening a second script from an existing script causes Nuke to crash out.
- BUG ID 3400 - [Perlin] Noise: misspelling of "lacunarity" parameter label (shows "lucanarity").
- BUG ID 3405 - Group/Gizmo: not updating from group.
- BUG ID 3410 - Out of memory from customer script with Primatte when degrain turned on.
- BUG ID 3412 - Animation menu Curve Editor option should open empty curve editor on un-keyed knobs.
- BUG ID 3418 - Nuke performance Issues on QuadCore with 3D.
- BUG ID 3429 - RGBA .pic files do not read correctly.
- BUG ID 3457 - File Browser: not recording -geo favourites correctly.
- BUG ID 3459 - Merge Materials node giving some strange results.
- BUG ID 3468 - Opening a Viewer on customer script crashes Nuke.
- BUG ID 3469 - Expression on node's 'disable' knob triggers a crash.

- BUG ID 3594 - Sampling colours for Primatte causes a double free or corruption crash under Ubuntu 6.06. This was due to differences in the versions of libstdc++ and libgcc_s on the build system (CentOS 4.5) and the customer system. Our Linux packages now include the libstdc++ and libgcc_s against which Nuke was built so that it can always find the exact same version.

- BUG ID 3598 - GridWarp crashes with no error message when trying to use blend slider.

- BUG ID 3605 - [x] expression no longer works. Fixed so [x] TCL command returns frame number except when it is executed inside a Colour > Math > Expression node.

- BUG ID 3610 - Script crashes when trying to disable Sphere node. It was trying to render primitives that should have been cleared when the Sphere was disabled.

- BUG ID 3618 - On rendering customer script with more than 5 lights in scene get - "double free or corruption" (Linux) or "Deallocation of a pointer not malloced" (OS X).

- BUG ID 3640 - Pressing Page Up in viewer causes crash.

- BUG ID 3657 - Tif Write is ignoring the premultiplied switch.

- BUG ID 3680 - Dilate node deadlocks.

- BUG ID 3728 - Write: Exporting to .mov fps is truncated to 2 decimal places. Increased scale factor to 1000 to retain 3 decimal places.

  Note that when a movie written with this version and fps of 29.9651 is loaded into Quicktime player on OS X, the info window displays fps as 29.96. However, checking value programatically shows correct result in file.

- BUG ID 3755 - Merge diagonal mode could zero pixels out due to loss of floating point precision with very small values.

- BUG ID 3778 - Nuke 4.7 viewer settings in script do not translate correctly into 4.8 - LUTs go to monitor, rather than viewer.

- BUG ID 3782 - 3D node's handle colour tints texture as well as colouring wireframe.

- BUG ID 3834 - Parameter linking in groups produces invalid expression.

- BUG ID 3850 - Crash when disabling MergeGeo nodes.

- BUG ID 3866 - OSX PPC - tifs written in 4.7v5 will not read into 4.8 (OSX Intel okay).

- BUG ID 3884 - Crash in 3D view on customer script.

- BUG ID 3931 - Fix for a crash with exr file with many layers.

Known Bugs & Workarounds

- On OS X, renaming the Nuke .app application directory from the command line makes Nuke unable to find it's startup script files when launched from the Finder (although it still starts up correctly when launched from a Terminal command line). Rename it through the Finder to avoid this problem.

Developer Notes

Changes

Here are the changes relevant to developers.

If you are writing C++ plugins, you will need to adapt your code to modifications that were required in order to support more than 64 channels. In order to do this, the long ChannelMask type has been replaced with a new object called a ChannelSet.

The ChannelSet can store a set of any number of channels. It also has the ability to store an "all" setting, and has efficient methods of testing sets for intersection and adding one set to another and iterating through all the channels.

Though attempts were made to preserve source code compatibility when possible, compatibility was not preserved in cases where type-unsafe practices were done, or where the compatible implementation would be particularly inefficient.

As ChannelSet is no longer a trivial-sized object, it was important for speed to use const references to it as arguments whenever possible. For this reason, ChannelMask is a typedef to const ChannelSet&, and thus function arguments that were ChannelMask are changed to the new reference automatically. This allows the majority of plugins to be compiled with as few changes to the source code as possible.

OLD: void MyClass::engine(int,int,int, ChannelMask channels, Row&)

NEW: void MyClass::engine(int,int,int, const ChannelSet& channels, Row&)

***Note**: This change is recommended but optional, as the old code will still compile due to ChannelMask typedef.*

There were a number of constants, such as Mask_RGB, that used to be constant ChannelMask values. These have been changed to their own type ChannelSetInit, which can be used anywhere a ChannelSet is wanted, and more efficient implementations are used in these cases. Avoid code that makes it not do the efficient implementation:

INEFFICIENT: ChannelSet A = Mask_RGB;

B += A;

EFFICIENT: B += Mask_RGB;

Changing a ChannelMask was done almost entirely by code that read "A=turn_on(A,B)". The new version only allows in-place modification by using the **+=** operator. This is much more efficient and easier to read in the vast majority of cases. Although non-in-place operators could be implemented, they were purposely not implemented to discourage their use.

OLD: A = turn_on(A,B);

NEW: A += B;

OLD: A = turn_on(B,C);

NEW: A = B; A += C;

There is an equivalent replacement for turn_off() which is to use **-=**.

OLD: A = turn_off(A,B);

NEW: A -= B;

Turn_off() was also often used in if expressions, so a version that returns a bool is provided for compatability. However, you may want to change it to the contains() method (be sure to swap the order of the arguments and invert the result):

OLD: if (!turn_off(A,B))

NEW: if (B.contains(A))

***Note**: This change is recommended but optional, as the old code will still compile.*

In Nuke, nearly all uses of intersect(A,B) with the & operator on ChannelMask were in the context of an if statement. As it is far faster to calculate this boolean result than the actual intersection, these were made to return bool. You have to use the &= operator to calculate an actual intersection:

OLD: if (intersect(A,B))

or

if (A & B)

NEW: (no changes needed, although A&B is preferred over intersect())

OLD: A = intersect(A,B);

NEW: A &= B;

OLD: A = intersect(B,C);

NEW: A = B; A &= C;

In a good deal of very old code, integers were used instead of the Mask_RGB enumerations. An "& int" operator that returns bool is provided for compatability, however the use of this is discouraged:

OLD: if (A & 2)

NEW: if (A & Mask_Green)

***Note**: This change is recommended but optional, as the old code will still compile.*

ChannelSet can no longer be initialized with an integer. The default constructor makes it empty:

OLD: ChannelMask foo(0);

NEW: ChannelSet foo(Mask_None);

or

ChannelSet foo;

You cannot assign an integer to a ChannelSet.

OLD: A = 1<<(z-1);

NEW: A += Channel(z);

Some code, mostly for debugging, cast the ChannelMask to an int. You can get the low 32 bits with the value() method. For obvious reasons, don't use this unless you know that only small channel numbers are used:

OLD: printf("mask=%x", (int)channels)

NEW: printf("mask=%x", channels.value())

The foreach() macro only works for ChannelSet. In other cases, you have to write your own loop:

OLD: foreach (z, Mask_RGB)

NEW: for (Channel z = Chan_Red; z <= Chan_Blue; incr(z))

***Note**: This works in older versions of Nuke as well.*

Plugins are no longer able to modify the ChannelMask passed to them, as it is const. They must copy it to a local ChannelSet variable and modify that.

OLD: MyClass::engine(y,x,r,ChannelMask channels,row) {

channels = turn_on(channels, Mask_RGB);

NEW: MyClass::engine(y,x,r,ChannelMask channels_in,row) {

ChannelSet channels = channels_in;

channels += Mask_RGB;

Be really careful if you are declaring ChannelMask local variables. Mostly, you want to change these to ChannelSet. However, if you are just making a local copy of another variable, it is more efficient to leave it a ChannelMask. Generally, incorrect use will not compile or will at least produce warnings from gcc.

OLD: MyClass {

ChannelMask foo;

NEW: MyClass {

ChannelSet foo;

OLD: ChannelMask x = Mask_Red;

NEW: ChannelSet x = Mask_Red;

***Warning**: It appears gcc will compile the old code without a warning, leaving x as a reference to a non-existent object.*

OLD: ChannelMask x = input0().channels();

NEW: (no change necessary as long as x is not modified)

OLD: ChannelMask myfunction() {return Mask_RGB;}

NEW: ChanneSet myfunction() {return Mask_RGB;}

Here, the old version fortunately produces a warning in most compilers.

OLD: ChannelMask myfunction() {return input0().channels();}

NEW: (no change necessary)

The old function mask(Channel) would turn a Channel into a ChannelMask constant. This is now done by automatic type casting, and mask(Channel) is a no-op. We recommend you delete it:

OLD: ChannelMask foo = mask(Chan_Red);

NEW: ChannelSet foo = Chan_Red;