Vixen Release Notes – 3.1.0

# New Features

### Main Application

* Profile Zip Wizard. This tool allows the user to select a profile, and all of the selected data and zip it in a file to the user’s choice destination folder.

### Sequence Editor

* Lip Sync Feature. This tool allows lighted animation elements to be described and sequenced by the user.
  + Supports single String, Pixel String, and Pixel Matrix Animations
  + Import existing Papagayo sequences
  + Lyric to Phoneme conversion tool - understands over 125,000 words.
  + User dictionary to allow user to define words not understood by Lyric converter.
  + Mark Manager / Tapper support for automatic phoneme sizing.
  + Export/Import facial bitmaps for use on pixel matrices.
  + Facial animations can be quickly remapped from string, to pixels, to Matrices and vice-versa.
* Auto save
* Added Preset Library. This provides quick access to your favorite colors, curves and gradients. Also adds drag and drop ability of these presets directly to effects on the timeline
* New Edit/Draw Mode.
  + Draw Mode allows you to simply click on an effect in the effect library and draw the effects into the timeline by clicking where you want it to start and dragging it to the point where you want it to stop.
  + Edit mode is the new name for the old mode where mouse clicks on the timeline effects will select them for editing.
* Edit Mode Improvements
  + Control-Drag to duplicate effects
  + Shift click to select multiple effects.
  + Improvements to Drag Selection box:
    - You can now use shift+drag to select additional "grouped" effects.
    - You can now use the selection box to select individual stacked elements on the same row.
  + New Drag Box Filter. You can now specify which types of effects will be selected by the drag box.
  + Improvements to Effect Snap
* Effect distribution alignment helper
* Multi Effect Editor - Changing only the intended values when editing multiple effects
  + Improvements to Add multiple effects dialog
  + New Indicator line when moving and resizing effects, makes it easier to line up the selected effect with other effects above and below it by adding a vertical line at the end that is being resized or both ends when moving an effect.
* Curve and Gradient Improvements
  + New Color Collections for assigning random/sequential colors to effects
  + Curve editor and Gradient editor are now accessible from the tool menu instead of only in the effects.
  + When items in the curve or gradient library are edited, the new settings update linked effects in the timeline right away without having to reopen the sequence.
  + Curve editor enhancements. Ability to invert or flip a curve and edit the point values directly. Handles for the curve points are more prominent and mouse pointer changes when over them.
* Mark Collections can now be exported
* Improvements to Undo / Redo
* Screen size and placement settings are now saved and restored when the editor is closed and reopened.
* Sequence export to FPP and CSV

### Execution Engine

* Significant performance improvements, especially when playing and sequencing in the editor.
* Unnecessary engine processes are skipped when the controllers are turned off to provide a better experience while sequencing.
* Live Context is only started when the Web Server is active and stopped when deactivated to save polling it if it is not in use.
* General memory consumption reductions.

### Display Setup

* When selecting Ok, the updated configuration is saved in the background rather than making the user wait at a hung UI.
* Stopping the controllers now actually stops them without a restart.

### Preview

* New Smart Objects
  + Icicles
  + Multi String
  + StarBurst
* New Distribute tools and property matching to the preview.
* New Alignment tools
* Zoom. You can now zoom your preview.
* Setup Previews Configuration pane saving happens in the background now instead of locking UI.

### Performance

* Editor should open significantly faster now especially for those with larger sequences.
* Editor Effect rendering speed has been improved for those with multi-core processors.
* Reduction of memory and CPU usage by the editor.
* Reduction of scheduled show startup time with faster pre-rendering on multi-core machines.

# Change Log

Much of the information in this change log is very technical or vague and will likely be meaningless to most users. However, there is a lot of detail here, including bug ticket numbers. So if you are looking to see if there was any work done on a specific issue, you may find it here.

## August, 2014

* Remove a GC.Collect that was causing the Nutcracker effect to block and prevent multi-threaded rendering.
* Updated Sequencer
* Make sure the fmodupdatetimer is not null is Dispose() to prevent an exception if it is null.
* The No-repeat code was preventing the RDS commands from reaching the RDS software.. I've limited it to be "Launcher" specific.
* Add global exception handler...
* Updated Version control with error handling that was causing V3 to crash in some instances
* Added additional error handling during parallel and thread based operations that could cause the application to crash if not handled properly.
* More Exception handling
* Change Curve to Default to ALL on to conform with current expections of Data
* Change default pulse level data to start solid and fade out.. so that chases start full intensity instead of other way around
* Fixed reference that was saving dll locally and causing other effects not to load
* Added Wipe from Inside Center out and Outside in
* No audio card and RDP issues fixed.
* Add the missing Cut operation to the graphical editor to support the function built into the shape window.
* This change is to help alleviate some object creation and ease the pressure on the GC.
* When a element has nothing going on it has no state. Instead of creating a new empty state object every time, let's use a static one across all the elements that doesn't have to be created every time and can be shared. This eliminates the new object overhead and then the subsequent GC collect an interval or two later when it is thrown away.
* The controllers themselves are individually threaded, and the process of updating the outputs individually is pretty fast, so the overhead of parallel updates on each output is not justified.
* Fix the button anchors on the instrumentation form so they stay at the bottom when it is re-sized.
* Add the missing Cut operation to the graphical editor to support the function built into the shape window.
* Add a way to clean up orphaned filters. Modified the Filter Manager
* to provide this functionality and then hooked it on the closing of the setup form. That way any orphaned Filters will be removed. This is a bit of a stop gap until the problem with the Graphical editor not being able to show them can be solved.
* Make a change to properly create the root node and children hierarchy
* to avoid putting orphan filters in the root chain until they are connected to something. This prevents filters that are not part of the full data flow from participating in updates. There is a further bug with creating and not being able to manage these unattached filters that needs to be addressed in another fix.
* Reorder some tasks to try and prevent a timing error with setting marks in the ruler.
* Move the default update interval back to a more normal 50ms
* instead of 46ms. This will give just a bit more time to process updates every cycle. Migrated to version 16 of the System config to update the users to 50ms if they are still at 46ms. If they have changed it to something else other than 46ms then leave it alone.
* Do some of the saving of the system config and module config in a background thread.
* Most of time this is just synchronizing the in memory state to disk, but the user has to sit and wait for that to occur. This frees the UI and user to do more productive stuff. Does not ease the pain of reloading the config on most cancel operations. That is a bigger nut to crack on another day.
* Another fix to the XML serializer to help the load time of larger configurations.
* Use a Dictionary for element looks as opposed to just iterating over a list thousands of times.
* Some changes to add some polish to the instrumentation.
  + Add a Close button to close the window.
  + Add a reset button to reset the values back to defaults and start over.
* Super tune the Context updates and Element updates.
* Changes include using the known updated elements from the context to drive how the elements update. We can avoid doing context look ups for state on elements that are not being touched by an effect this update and just set them directly to empty like would have happened after all the look ups. This overall looks to reduce the time spent in context and element updates by about 30-40% in some simple timing based benchmarking.
* Combines my DragBoxFilter, UndoAddMultiple & DrawModeWork & ColorCollections branches to make merging smoother by resolving conflicts
  + Adds a drag selection box filter
  + Adds Drawing mode which allows effects to be drawn onto the Grid
  + Adds a Color Collection Library, which allows random or sequential assignment of colors to effects.
  + Adds ability to selected individual stacked effects on the same row and time period.
  + Adds ability to select additional effects with the selection box, by holding shift.
* Updates to Add Multiple dialog:
* Fixed to work with undo engine
* Added options to alight effects to beat marks.
  + Removed stray ; that would not allow playback to be resumed with space bar
  + Fixed bug when resizing effects if control was pressed (element could become null) Another nice speed up in loading the editor for large element counts.
* Defer adding controls to the parent control for row labels until the row is actually visible. Adding controls is very slow and when you are adding 1000's of them it can take a long while.
* Some simple code cleanups. Remove unused variables, make a few constants, Linq simplifications.
* Changed tooltip for the icicles button.
* Significant speed up in loading sequences into the editor with large node/effect counts.
* Only create the node dictionary once and reuse it rather than do it for every one we need to add.
* Fix clone methods so they work, as they are used in some editor operations. Issue reported with VIX-347 that is caused by this bug.
* Add help links for the new icicle preview object.
* Some cleanup to make the preview more efficient. It was wasting time looking at all the intensity values, but in reality all we need to do is see if there are any states in the collection.
* Fix missing title bar text when sequence loads.
* Added icicles to the preview.
* Make a change so that when you disable the controller in the setup page that it actually stops the hardware update thread that is associated with it. Also bypass the filter chain if we do not have any controllers running to save CPU cycles.
* VIX-392 Direct comparison of floating point numbers with a bit of rounding error leads to them never matching for Blue and Green. Change the comparison to allow for a bit of tolerance.

## July, 2014

* Removed duplicate preview files.
* Fixed a couple preview shapes (Triangle, Arch), fixed some issues with the preview main window. Arches were not showing up in the preview in v11. This work is for VIX-471 & VIX-472
* Adds Drawing mode to sequence editor, which allows users to draw effects into the grid vs drag and drop. Users can select this mode as an option.
* Adds indicator lines to effects that extend top to bottom of the grid when resizing or dragging an effect. This is an option on the Edit menu, user can select color.
* Also removes a stray ; in the if statement for resuming play in the editor using the space bar, that would cause pass thru of else and stop the sequence
* Only calculate the curve points once rather than for each node.
* Fixed a problem highlighting pixel #1 in a string that has no first pixel. Was causing an exception if there were no "pixels"
* Something happened to the preview menu. It is back. Re-did the previously lost fixes to the preview.
* VIX-460 Initialize Bottom Left Point during Grid setup to prevent Null Pointer and InvalidOperation Exceptions
* Clean-up of previous commit.
  + Makes better use of available methods
  + The Dialog now determines how many effects are possible for addition and displays this in a label at the top of the form. The form caption also shows which effect is being worked with.
  + If something goes wrong with calculations the user is prompted with a Yes/No continue anyway option. This takes place once the user clicks OK on the dialog.
* Additional work for VIX-433. Fix a bug that would not properly track and undo a move of an effect between rows.
* VIX-355 Added path separator character to separate Vixen 2.x pathname and parsed filename in conversion routine
* Corrected output path for Helix controller
* Adds ability to select which effect types will be selected by dragging a selection box also affects Shift-Click selections
  + This commit also removes old code that was commented out, that dealt with effects on the toolStripMenu
* Fix the default curve to start at 0 and go to 100 so the pulse does not have to band aid it up to the same with extra intents.
* VIX-364. Make the node handles filled and stay on top in the corners so they are easier to see. Make them easier to select when they are along the edge. Change the cursor to a cross hair when over the node handle to better indicate it can be dragged.
* This work modifies the code that adds multiple effects from the context menu, so that it is wrapped into one action by the undo/redo engine

## July, 2014

* Helix: don't copy NLog to output directory
* LipSyncApp: reference Controls project properly.
* Don't copy NLog to App module output dir.
* Squashed merge of James B's EffectsAddMultiple branch. Untested, conflicts manually resolved.
* Merge Jeff's UndoRedo branch.
* Merge Jeff's SnapFix branch.
* Squashed commit of Ed's Foghorn / master branch:
  + VIX-463 - Aborting rename to Mark Collection causes name to be erased. Added null check to after label edit function.
  + Added check in Text Converter Dialog code to protect against the possibility of null Mark Collection Names.
  + This protects against issue reported in VIX-463
  + VIX-197 - Corrected compile errors due to code merge
  + VIX-444 - Unable to set change interval to less than 100ms in Alternating Effect
  + Modified AlternatingEffectEditorControl.cs - trackBarInterval.Minimum property to have a value of 1 instead of 100
* Additional work for VIX-433. Refactor the context menu alignment tools code. Moved and reworked the methods into the grid to provide support to group the effects together as a complete operation when adjusting their times. This way it can be added as a single entry into the Undo/Redo queue instead of many single moves for each element manipulated. Added 2 new move types of Align and Distribute to put better descriptions on the undo operations listed.
* Code clean-up for consistency.
* Fix a bug that would leave the last clicked grid location in a bad state when elements are moved in a certain way and then the next step is to immediately shift drag lock. Thanks to James for finding this.

## June, 2014

* Fix up some threading issues that could cause an exception to occur because a menu item was being manipulated on the wrong thread. Also fixed a misspelled method name in the same space.
* Fix a bug where the calculated element snap points were being added to the the static snap points when an effect was lined up on a mark. This caused the element snap points to be left as static if the effect was moved off the mark. Only marks are considered static. This bug was additive on every move or re-size, so over time could have slowed things down and used extra memory in the editor.
* Comment out non-building code, for Ed to have a look at.
* Remove the Debug targets from the Vixen solution.
* Fix up the LipSync project files to put artefacts in the correct output directories, and also remove the Debug targets.
* Merge ebrady1's Lipsync work. Untested, not compiled
* squashed merge point only.
* Bug fix for related to commit a7cc6df and VIX-343 to make sure the snap times for Marks are recalculated when the snap strength changes. Also make sure it is applied to the marks in the ruler.
* A couple of minor performance tweaks. For each over enumerator rather than convert to list to do a linq expression. Convert initializer to use member variable to prevent unneeded events firing.
* Usability enhancement. Now that the Curve and Gradient Libraries can be accessed from the Editor toolbar, the user instinct is to double click the Curve or Gradient to edit them. When used from the Effect editor double click is a select. Enhance both library classes to specify how the double click behavior acts, so they can be used in two different contexts.
* Clean up some messy code that populating the Audio drop down and the waveform. It was creating a FMod Instance twice. Refactor to only do it once.
* VIX-343 Implement feature request for adjustable snap strength. Added 4 levels of strength in a drop down menu alongside the Enable/Disable SnapTo button on the tool bar.
* Fix a bug that would not properly restore the Snap to enable/disable setting.
* VIX-357 and VIX 368. Detect if any audio devices are available and if not disable the audio so it does not crash.
* VIX-349 - Fixed set level issue.
* VIX 62 and VIX 434. Change the SequenceExecutor to allow data to be inserted while playing. I have tested this quite a bit I have seen no ill effects. It is easy enough to reverse if something comes up with broader testing.
* Code clean up in some core editor classes.
  + Remove unused using statements.
  + Remove unneeded qualifications including this.
  + Remove some dead code.
  + Tweak a few error messages. Ensure errors are being logged correctly. Some of this has been bugging me for a while, so I just decided to clean it up a bit.
* Fix a bug that occurred when moving or resizing elements where the rowelement collection was not reordered when a element has moved or re-sized that caused the overlapping element logic to miss elements that become overlapping.
* VIX-450 Feature request to allow Shift click selection of elements.
* Force the setup form back on top of a newly started preview. The preview launching in front of the form was annoying. This fix is probably not the prettiest, but it is simple and it works.
* Add Color Gradient Library editor access to the Tools menu to go along with the Curve Library Editor. With the work on VIX-68 this will give central access to the editors without having to edit an effect that has the library curve or gradient in it.
* VIX-68. The editor is now aware of curves and gradients that change and will re-render any effects tied them in the background when they change. Required some changes to the effects to properly update the IsDirty flag.
* Fix a rendering bug when resizing elements.
* Part of VIX-433. Fixes cut or paste operations to be a single undo event instead of making individual entries for each effect added or removed. This way a cut or a paste operation can be undone as a whole.
* Fix the Clone method so it works.
* VIX-347 Implement feature request to Ctrl Click and drag to duplicate effects. It works with a single effect or if multiple are selected it will duplicate and drag all of them.
* Fix a bug introduced in commit 86c3908 with the settings work that would allow Time per pixel to be negative when creating a new sequence.
* Change contributed by James Bolding to lock the X or Y axis when dragging effects or during the drag select using the Shift or Alt + Shift keys.

## May, 2014

* Additional work for VIX-340. Add support to restore the window to maximized if it was closed when maximized. Add some sanity checks to make sure it is on the screen.
* VIX-315 Fix a bug where the effect info message would be cut off if the effect was near the top of the grid. Now the info message will be shown below the effect if there is not enough room above it.
* Partial work on VIX-340 and VIX-455. Consolidate and organize the existing settings into the settings.xml. Clean up and enhance the XMLProfileSettings class to better handle and organize settings into Profile, AppSettings, and Preferences. This will also allow the editor to save it's window positions and some other toggle settings. IT will allow for more settings to be persisted. It will also be the ground work for a new Preferences dialog that any module can contribute a preference pane to and set preferences to be retrieved.
* Changes for VIX-348. Created an auto save menu item under Sequence that allows auto save to be turned on or off. Interval is set to 5 minutes for now. Need to create a general configuration dialog where items such as the save interval can be configured. Will create another enhancement bug for that work as it is bigger than this item. Auto save on/off setting will be saved and restored over editor sessions. Interval is also saved in preparation for configuration dialog. Attributes are saved in the Settings.xml in the user profile that is already purposed.
* Changes for VIX-341. Rework the looping function in the Editor to remove the loop timers and associated logic. Enhance the SequenceExecutor to handle looping so the delay can be eliminated. Make the editor use the new operations. Re-enable the ability to pause while looping.
* Minor performance tweaks.
* Minor change to use a linq structure.

## March, 2014

* Update text message to make it nicer.
* Fix duration check to allow for durations down to 1 millisecond. This allows for the work on VIX-444.

## February, 2014

* Fix VIX-359. Fix the waveform drawing routines to allow for sequences longer than the audio duration.
* Fix VIX-401.
* A few more memory clean ups.
* Put back some global logging messages that where lost in the big merge to mainline.
* Fix overreaching constraint on element move.
* Also restrict another check to not allow start time equal to the end time.
* Merge the support-report packaging wizard from bolwire.
* Preview zoom repairs... someone should really things before they commit them!
* Preview zoom with the mouse wheel.
* Preview zoom on the status bar with a tracker.
* Added zoom to the preview (Ctrl+ or Ctrl- only for now, mouse coming soon)
* Fixed an error with the arch matching (caused a fault) Finished the candy cane matching.
* Added distribute tools and property matching to the preview.
* Added alignment tools to preview. (left, right, top, bottom, horizontal, vertical)
* VIX-317: Corrected the bug that keeps a the background file open when assigning to different preview windows.
  + Added a lock to the preview image resize dialog box to maintain the aspect ratio when resizing the image. There is also an option now to resize without maintaining the aspect ratio.
  + Updated rotate Star and Starburst.
  + Added XY rotation to the Starburst.
  + Cleaned up a bunch of comments in the preview shapes.
  + Added a starburst to the preview and the consolidated setup screen.
  + Some fixes to the new Star Burst preview prop.
  + Added a "Star Burst" to the preview and removed VixenPreviewSetup which was very old and unused.
* Updated Version control with error handling that was causing V3 to crash in some instances
* Add global exception handler.
* The No-repeat code was preventing the RDS commands from reaching the RDS software. I've limited it to be "Launcher" specific.
* Make sure the fmodupdatetimer is not null is Dispose() to prevent an exception if it is null.
* Change the preview to use alpha-blended colors to give a slightly nicer look.
* Added more logging for the BlinkyLinky module.
* Fix small artifact in the border that allowed a underlying pixel to stick out.
* Fix a bug that occurred when undoing an effect re-size. It was not really related to the effect drawing routine changes, but existed because the effect was not re-rendered after undoing the re-size.
* Fix a drawing bug when resizing effects.
* Updated distribution dialog to use masked textbox for time inputs, and validation of format and minimum effect time.
* Fix effect element border drawing.
* Merge branch 'master' into EditorRevamp
* Add some additional checks to prevent effect elements from getting into bad time state. Part of VIX-318.
  + Only iterate up to the last visible snap point.
  + They are in time order, so no need to iterate further than necessary.
* Cosmetic changes to clean up the various different naming conventions into one standard.
* Restore clipping rectangle.
* Completed work on exporting beat mark collections.
  + Collections can be exported in audacity format, or in Vixen 3 format.
  + Vixen 3 format retains all information about the collection, including colors & weights.
  + The default Vixen 3 mark extension is .v3m
* Add caching of the smaller visible portions of the elements.
* The cache will be invalidated if more of less of the effect is visible, but this makes a noticeable improvement during playback when the grid is repainted on every cursor tick.
* Stop unnecessary re-painting of the waveform.
* Only set modified/unmodified state if it is not already set.
* Tweak element stacking routine to be a bit more efficient.
* Addition of effect distribution alignment helper
* Addition of align center points helper
* First cut at on the fly effect image rasterizing. Images are no longer cached. They are built on the fly. This reduces memory consumption a considerable amount. Includes some other cleanups and code reduction associated with unneeded functions. Speed improvements in the painting routines.
* Branch for working on beat mark exporting

## January 2014

* Global error trapping and logging.
* Add the curve editor to the main tool bar editor.
* Add reverse and invert features as well as the ability to directly set the numerical coordinates of a point to the Curve editor. Completes VIX-329.

## December 2013

* Added check to make sure destination direction exists, and if not the option to create it before proceeding.
* Added check to CompressFolder() to make sure source folder exists, and if not return. This would most likely happen if a user has renamed one of the default folders in their DataFolder, this prevents CompressFolder() from throwing an exception if it can't find the specified folder.
* Changed Create Zip File button to Ok, and aligned beside Cancel button to look more like a standard dialog window.
* Modified code to always pull log files from MyDocuments\Vixen 3\Logs
* Pressing Return(Enter) on the form invokes the OK button.
* Added status bar to indicate work is being done.
* Add a Profile Zip wizard This tool allows the user to select a profile, and all, or the selected data and zip it in a file to the users choice destination folder. When everything, or application data is selected, the AppData folder will also be included, and in the .zip file under the Vixen folder. The Zip wizard is accessible from the profile editor window.
* Added two items to context alignment menu, Start to End, and End to Start. Also moved Alignment tools to Alignment flyout from context.
* Fixed small issues with Align Start/End times where the element could become invalid
* Added check for Shift modifier key when usings Match Duration, when shift is pressed the duration match is done, while holding the effects end time, by default the match is made while holding the effects start time. This is also noted in the .ToolTip for the Match Duration menu item for the user.
* Updated Sequencer
  + Fixed spacebar not stopping sequence when using loop, it would just restart the loop.
  + Interpolate HSV colors properly -- wrap around the 360 degree point on the color wheel, rather than doing all transitions within the 0 -> 360 bound

# Previous Version Release Notes

## == 3.0.10 ============================

### New features:

- new single setup form for display configuration! No more setting up displays in 3 parts -- elements, controllers, patching -- it's all done in one page!

- added virtual Helix controller for exporting data

- added K8055 controller

- extra mark manipulation in the UI: can select marks in the ruler, delete them with keyboard shortcuts

- extra sequencer UI improvements: shortcuts, right-click context menu, disable snapping

- added version control module to track files

- improved instrumentation and performance monitoring

- improvements to the E131 controller, to allow customization of how data is re-transmitted and/or buffered

- added looping in the sequence editor

- added delayed playback in the sequence editor

### Fixups:

- Improvements to serialization if there is 'bad' data in the config files

- many small improvements/fixups to the preview

- many small improvements/fixups to the nutcracker effect

- performance improvements

## == 3.0.9 ============================

### New features:

- a new and improved Scheduler and Show module! See the forums and videos for more info.

- More performance improvements to reduce CPU usage and memory usage.

- RDS module improvements: supports the new VAST application, and other improvements

- Added the Elexol USB I/O controller

- Vixen 2 sequence importing: can now repeat elements when dragging to the mapping window (eg. for multiple color mappings).

- Sequencer UI improvements: Context menu & options

- Sequencer UI improvements: Dockable windows (with a vertical effect list)

- Sequencer UI improvements: marks in the main window, and mark operations in the ruler

- Split the application to have separate 32-bit and 64-bit versions, to allow for larger memory usage on high-end machines

### Fixups:

- The preview was not opening at the correct size sometimes, or with a correct background; now fixed.

- Fixes for importing audacity marks

- Fixes issues with 'lost' elements in the config

- Fixes with preview rendering of discrete colored elements

- Fixes to output a more accurate value (eg. 8-bit value) when converting a color to a value

- Fixed an issue where autoplay could leave some elements "hung" or not turning off

- Fixed an issue with V2 sequence imports where it would read in too much data and try to map to non-existant elements.

## == 3.0.8 ============================

### New features:

- A ridiculous amount of speedups to the entire application: UI responsiveness, rendering/drawing speed in the sequencer, preview performance speed, and so on. For example, effect rendering is now performed in the background in another thread. The application is now much more responsive and usable.

- A similiarly ridiculous amount of improvements to memory usage; particularly where nutcracker effects are used.

- Added a 'webserver' module, to allow basic control via a web interface.

- Added support for "Elexol" controllers.

- Added an RDS and Launcher control modules.

- Added support for a Direct 2D rendering mode for the preview (which has been temporarily disabled to ensure stability for all users, until we can get more testing performed)

- Changed logging system to use NLog, instead of a custom homebrew system

- Added support to be able to pick a specific audio output device in the sequencer

### Fixups:

- better handing of default values for all effects

- many minor preview fixups

- Some nutcracker effects would render 'backwards'

- Some preview items (eg. nets) would not draw/render properly

- The positioning/order of some preview shapes (eg. squares around windows) has been improved

- fixed some crashes when opening/closing sequences at unconventional times (during playback, paused, etc.)

## == 3.0.7 ============================

### New features:

- Discrete color support. There is now a 'color' property that can be added to an element, to define how color

should function for that element. Each element can be configured in three ways:

\* Single color

\* Multiple independent (discrete) colors

\* full color (eg. a mixing RGB light)

(If not configured with the color property, an element is assumed to be full color.)

If single or discrete colors are selected and configured, when configuring an effect against that element, only

the available colors will be visible to be used.

- Color breakdown filter: added 'mixing' option, to complement the discrete color property (above). If enabled,

the input color will be 'mixed' by using proportions of the configured colors. This would be for items like RGB

pixels or floods, which can 'mix' colors. If the option is disabled, a color will only be output from the filter

if it matches the input color EXACTLY. This is intended for operation with discrete (or single) color items.

For example, if you had a mixing RGB color breakdown filter, and the input color was purple, then the filter would

output proportions of blue and red to get the desired color. However, if it was a non-mixing RGB color filter, it

only output colors that are Red, Green, or Blue exactly. (Presumably, you have a tree with R/G/B strings on it,

so only want those exact colors.)

- Better support for importing Vixen 2.x sequences! Will now support making 'maps' for 2.x channels to 3.x elements.

These can be saved and reused for future importing of sequences. Also supports color: you can now specify the color

of the incoming data/effects, so the V3 data will be generated with effects of the appropriate color.

- New 'wipe' effect, and associated 'position' property for elements (to define a location in the display).

- More Nutcracker improvements and effects: 'Movie' effect, speed improvements

- Sequencer responsiveness and speed improvements -- backgroun rendering, speed fixes, memory fixes, etc.

- New 'alternating' effect (thanks Gizmohd!)

- easy setup for element 'templates' (pixel grid, and megatree) -- right click on the tree in the Element Setup

- starting to add icons for effects in the sequencer

- audacity support in the mark manager

- added framework needed for launcher output and RDS output plugins (not yet complete)

- added a 'debug' controller to see specific values/commands that are going to the controller

### fixups/bugs:

- UI 'flow' fixups in sequence editor

- display preview shouldn't hang when created on some (unlucky) machines (thanks to dlovely for help debugging!)

- fixed bug where effects might 'stack up' when pausing and restarting while an effect is displayed

- information about effects in the sequencer are now displayed when hovering over an effect, not permanently

- efficiency improvements for rendering of spin, chase and nutcracker effects: (not generating useless data, etc.)

- E1.31 module fixups, to turn off channels properly

- Element Tree in setup: now more efficient when rendering, and won't hang when grouping hundreds of elements

- Fixed DLL references for output modules that use the parallel port (Hill320, 595)

.... and possibly more that we've forgotten!

## == 3.0.6 ============================

- Display preview! There's now a new display preview module, written by Derek Backus. Still in development, so feel free to give feedback.

- Nutcracker effects! Derek's also been hard at work on Nutcracker effects, which are fresh off the blocks. Still in development; feedback welcome.

- Huge performance improvements, for application responsiveness, UI improvements, etc. Still more to go...

- Controllers: 595 and Hill320 fixups to get working.

- Sequencer: mono audio files now work.

- Sequencer: audio detection/frequency detection implemented.

- Admin: setup dialogs won't prompt to save changes if nothing has changed.

- There's now an installer for the application.

## == 3.0.5 ============================

- Importer: A Vixen 2.x sequence importer has been created and added; you can now open \*.vix files. Improvements still need

to be made to the mapping to 3.x elements and colors, but the basic concepts are there.

- General: A new Vixen 3 logo and application icons have been created!

- General: the term "Channel" has been renamed to "Element" (to describe a logical part of a display, such as a

minitree, prop, etc.). The term "Channel" is now used for the outputs of a controller; ie. to bring it

in line with how the community uses the term.

- General: Corrupt or empty system config files no longer cause errors on startup; they get moved aside, and a blank systemconfig loaded.

- Controllers: A FireGod Dimmer control module has been added. Still needs user testing.

- Controllers: All serial controllers (Renard, Generic, Firegod) have improved serial port handling

(retrying open serial ports, IN USE markers, etc.)

- Display Preview: preview window can't be accidentally closed (the instance must be disabled in the Admin panel instead)

- Sequencer: A pseudo-grid has been added. To add a grid to a sequence, go to the Mark Manager, add a new mark collection, and

use the "Generate Grid" button. It will ask for the time period (with 50ms as default), and generate regular

marks to correspond to that time period. Effects should snap to the new marks.

- Sequencer: opening a saved sequence with audio properly draws the waveform on open

- Sequencer: adding audio to a sequence that doesn't have it doesn't get the rows out of alignment

- Sequencer: audio waveform doesn't block the UI while it's rendering, and is much more usable (especially while zooming)

- Sequencer: Twinkle effect: can't set the coverage to 0% anymore (now a 1% minimum, as 0% causes errors)

- Sequencer: Fixed a spurious error dialog (MouseUp in MouseState) that was accidentally getting called.

- Sequencer: click-and-drag scrolling in the sequencer now works when the sequencer is hard on the right side of the monitor

- Sequencer: The pause button now works properly, and doesn't cause errors when trying to resume

- Sequencer: Effects: some effects have had a "depth" setting added, to control how many layers deep the effect should apply.

- Admin: The setup dialogs will now warn if you cancel out of them (ie. losing changes).

- Admin: Elements & Groups Form: Fixed an issue when when dragging to copy and paste with the mouse (last element didn't copy).

## == 3.0.4 ============================

(Versioning has now dropped the 4th digit.)

- When serializing sequences, module data that is for non-existant modules is now discarded.

- Display preview update increased to 50ms from 10ms. Not currently customizable.

- Scheduler improvements to cache programs and prerender all sequences.

- updated OpenDMX controller module.

- Administrative cleanups of projects, namespaces, public repository, etc.

- Fixed up a potential crash in the Chase effect.

- Added Generic Serial controller module. (macebobo)

- Added a Candy Cane shape to the display preview (alphageek)

- Removed the "Generate Channels" button from the controller form; it encouraged bad configuration.

## == 3.0.3.1 ============================

VIX-135: Locking the instance collection in ReleaseContexts when clearing it.

VIX-111, VIX-138: Making the data directory portable.

VIX-133: Color Filter mapping now working and copy does not work.

Fixed bug where updating chained filters may throw a null reference on startup.

VIX-138: Better null handling in the channel manager for a lookup.

Updated the audio media module to support variable speeds.

Updated TimedSequenceEditor to support changing the speed of the timing source, when the timing source supports it.

Changed where IsRunning is being set to true in SequenceExecutor.cs so that the media will have been started by that time.

ColorBreakdown Filter: added extra template options from forum requests (RGxB, and GRBW).

Filter & Patching setup form: Added a "high quality rendering" checkbox. Doesn't make much difference, but might help on high shape counts.

VixenApplication: fixed issue when creating filters through the wizard where the module data wasn't being duplicated properly.

Added DimmingCurve post filter.

Added OpenDMX module, care of chilloutdocdoc and piense on the DIYC forums. (thanks guys)

DisplayPreview: added error handling for missing background images.

VixenApplication: improved load time for the filter setup form when there's \*heaps\* of shapes to draw.

VIX-137: DisplayPreview: better null handling on channel updates.

Twinkle: don't make data for channels that have already had data generated for.

Chase: Fixed error with rendering, potentially throwing an exception.

Timeline Control Grid: fixed a few rendering issues/hangs/etc. Pretty messy and hacky, but I think it's time for a control rewrite next year. This at least stops the hangs.

Chase: render more finely, to pick up quick chases.

Twinkle: added another color option to spread the color across the subchannels.

TimedSequenceEditor: added negative time support in the mark manager.

TimelineControl Grid: dodgy hack to avoid any unhandled hangs from the drawing problems. Temporary fix.

ColorGradients & Curves: added extra shortcut keys to editors, and persist the library selector window size/position.

TimedSequenceEditor: add support for editing multiple elements at once (only of the same type).

Curves: add null library check, to avoid potential nasty bug on shutdown: if the curves library module is closed before another module that is using it (eg dimming curve), it can barf pretty hard.

Added a "data\_dir" command-line argument for overriding the root data directory and moved data loading to before the initialization of the framework.

Use:

data\_dir=c:\path\where\you\want\data

data\_dir="c:\path where you want data"

Notes:

1. The VixenApplication app will default back to the default data directory if it cannot find its data file within this path.

2. This path is \*the\* data path for both the application and the framework. This means that if it's a new path for the framework, it's going to build out its directory hierarchy.

3. Do not put spaces around the equals sign! If the data directory path does have spaces, you can put quotes around the path, but around the path ONLY. Example: data\_dir="c:\vixen data"

## == 3.0.3.0 ============================

Preview improvements and fixes.

E1.31 module included. Untested, but claimed to work.

"Simple" scheduler module included. Sequence execution tested, program management and execution not tested. Still subject to future changes.

Lots of fixes to the timed sequence editor.

Lots of fixes to the channel config form.

Interpolator precision fix.

LightingValue now makes use of the alpha channel.

## == 3.0.2.2 ============================

Fix for loading a sequence from a non-qualified file path.

Moved user script initialization outside of the script's execution thread so that execution is considered "started" when Play is actually called.

Minor fixes to error handling and null-checking when stopping the system with a busted config.

Fixed issue when loading Preview modules: was looking for the wrong attributes.

When playing from an effect buffer, reset the last read/write counters when starting fresh.

Fixed sequence data sort issue.

Audio waveform visual in the timed sequence editor.

## == 3.0.2.1 ============================

Added dataflow component circular dependency checking.

Added naming templates and generators.

Module store serialization fixes.

Refactoring of system file migration.

A plethora of patching UI fixes.

Fixes for SetLevel effect.

Timed sequence-type fixes.

Timed sequence serialization fixes.

The beginning of a patching wizard...feel free to provide feedback.

## == 3.0.2.0 ============================

Data handling changes. HUGE!

See http://doityourselfchristmas.com/forums/showthread.php?22100-September-2012-3-0-2-0

New configuration dialog in the admin app to accommodate filters and data flow.

Output device refactoring. They are mostly internal, but there are some external changes. All modules in the git repo have been updated.

Added a ValueTypes common assembly for commonly-used types of data, such as percentages and file paths. This is to prevent modules from redefining similar kinds of data in different ways and having to create editors specifically for themselves. An editor for any of these common types (there are only two so far) can be used for any effect that uses that type.

Added an Index member to outputs. This is set when the output is instantiated and is not modifiable.

Added VixenSystem.Filters for the new filter layer.

Added VixenSystem.DataFlow for managing the data flow from channel to output through the filter layer.

Added VixenSystem.OutputControllers, VixenSystem.SmartOutputControllers, and VixenSystem.Previews for specific handling of those device types.

Added VixenSystem.ControllerManagement and VixenSystem.OutputDeviceManagement for generalized handling of those device types. This allows for operations such as VixenSystem.ControllerManagement.StartAll() which affects both OutputControllers and SmartOutputControllers.

Previews now receive intent states instead of commands.

Added non-static data store to ModuleStore.xml.

System-level module data (such as for controllers) has been pulled out of SystemConfig.xml and put into ModuleStore.xml.

Fixed CPU usage statistic in the admin app.

## == 3.0.1.8 ============================

Handling BadImageFormatException explicitly to allow non-assembly DLLs to reside in the module directories.

Moved some non-assembly libraries to the application root directory because that's where they have to be to work.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Modules have been split from the core source repository at this point.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## == 3.0.1.7 ============================

Changes for sequence and language modules.

## == 3.0.1.6 ============================

How come there hasn't been an update in so long? Well...

Removed the "Runtime Behavior" module type. They will be doing their phoenix impression at a later date.

The "Sequence" module type is now the "SequenceType" module type. This is a significant change, so it's detailed below. \* THIS IS A BREAKING CHANGE. \*

New SequenceContext and some changes to how execution contexts are requested and fulfilled.

Slightly revamped data sourcing model for contexts.

Existing timed and script sequence types have been reimplemented as modules.

Expansion of service classes to better serve up core services through Vixen.Services. :)

(Okay, okay...this is the start of a consolidation effort to put all available consumer services in one easy-to-find place.)

\* Sequence => Sequence Type Module Change: \*

This removes any concrete definition of a sequence from the core assembly and puts it entirely into modules. The core assembly defines a sequence interface, as well as other interfaces necessary to the sequence type module, but there is no sequence class defined. That's the job of the module. This means:

GOOD: Sequences can be defined more freely now, within certain limits defined by the interface (e.g. still have to work with effect data, sequence filters).

MAYBE SCARY: The module is now responsible for everything having to do with that sequence type.

Since the module now defines the sequence class, it also needs to define several other things:

1. How that sequence will be stored.

The sequence-type module's data class will be used to define how a sequence will be stored. There is a base class to handle the mundane members of the ISequence interface, but it's up to the module to extend it to define any additional members.

2. How older sequence files of that type will be brought up to date.

Versioning is a requirement of all files. When a sequence is opened and it's not of the latest version, the module needs to handle bringing it up through intervening versions to make that sequence current.

3. The biggy: How the sequence is executed.

It's through this that everything is told that a sequence is starting, that it's ended, what rules its timing...everything.

NOTE:

Like we said, it may be weeks between updates because of large-granular changes like this that are pretty core changes. Not that we aim to do core changes at this point, but a conceived real-life use demanded that it be attempted.

## == 3.0.1.5 ============================

Brought back effect preview rendering in the editor, but only for effects that use LightingValue commands. So far, that's all of the ones that are in the package.

## == 3.0.1.4 ============================

Added abstraction for output device update signaling.

## == 3.0.1.3 ============================

Removed Write() from Vixen.Execution. Replaced it with a formalized "live" context available by calling VixenSystem.Contexts.GetSystemLiveContext().

## == 3.0.1.2 ============================

Changed the data directory to "Vixen 3" to avoid clashing with 2.x.

Removed TestPreview.dll from the build.

## == 3.0.1.1 ============================

Spin and twinkle effect updates so they will work with the new changes.