quadrium2
User's Guide

About quadrium 1

How it works 2

quadrium Window 9

Starting Points Browser 10

Drop Targets 12

quadrium Documents 13

Rulers 15

Snapshot Panel 15

Details Panel 17

Working with Structure - Nodes and Wiring 19

Editing Node Parameters and Settings 23

Mutations 25

Tweening 27

Creating Movies 29

Document Information 31

Breeding 33

Working with Gradients 35

Menus 37

File Actions 42

Preferences 45

Rendering 45

Local Sharing 46

.Mac preferences 46

Behavior 47

File Actions 48

AppleScripting quadrium 49

To create a new node in the document: 49

To “wire” something up: *50*
To set a parameter: *50*
To change a setting: *50*
Finding nodes in documents: *51*
Manipulating the current point of view: *51*
Common Document Operations: *51*
Additional Actions *52*

About quadrium

quadrium is an application allows you to create a wide variety of abstract images, textures, and animations. It combines a wide variety of mathematic techniques (including fractals) with an interface designed to let you easily explore the vast realm of possible images, allowing you to literally “grow” you image. You don’t need an advanced degree in math to be able to create rich and fascinating images, suitable for websites, presentation backgrounds, cards, CD covers, brochures, or fine art prints. You can even create animations, bringing your creations to life.

At it’s heart, quadrium creates a mathematical function that translates the coordinates on the screen into colors. These functions can vary from simplistic (to make a simple gradient, for example) to complex fractal generation, and a whole variety of different functions that are unique to quadrium. quadrium is also designed to be fast - live feedback is one of the most important usability features, so you don’t have to wait to see what happens when you change something - the image updates as you change the settings. quadrium makes extensive use of the Altivec velocity engine on PPC machines, and SSE2 extensions on Intel based machines.

All of this complexity can be overwhelming, so quadrium uses techniques of “genetic algorithms” to allow you to evolve your creation. Just as plants or animals are selectively bred to enhance specific traits and remove others, so images in quadrium can be breed to enhance specific features and remove others. quadrium also allows you to see simpler variations of the current image - and since it automatically tweaks the wide variety of parameters that make up an image, you can see the results, rather than trying to figure out how to make them happen.

The interface is further refined to encourage experimentation. Temporary results can be saved for later exploration, and intermediate results and variations are saved with your image. Your work can then be shared locally, allowing others on the same LAN to see your work, or globally via your iDisk (.Mac account required to share your images with others).

The results can then be rendered to a high resolution file or printed (you can also things up to have the file automatically added to iPhoto, or create a custom Automator workflow or AppleScript script to be executed, allowing you to automate the process of uploading your images to a website, for example). You can also use variations on your image to become key frames in an animation, and generate a movie that morphs from one to the next.

How it works

quadrium works by generating a formula that can convert image coordinates into colors. For example, one could make a formula that calculates the distances from the center of the image, and uses that value as a grayscale value. The result would be something that looks like this:

Points located near the center of the image (and thus having a distance of 0) end up black (their red/green/blue components are 0,0,0), while further away they white. If you want to write that formula down, it would look like:

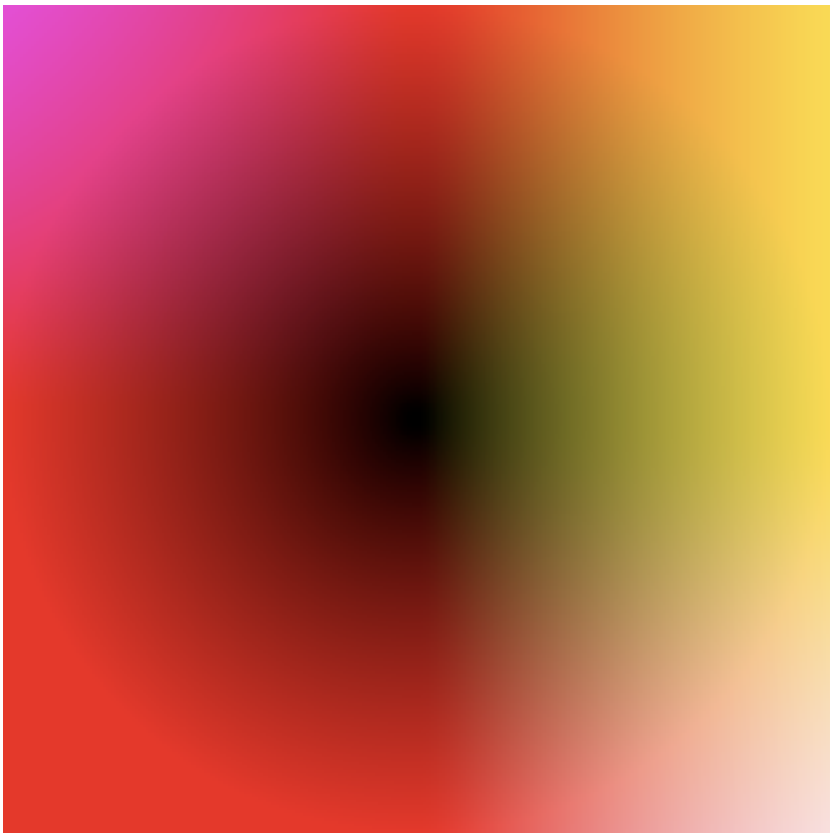
sqrt(x * x + y * y), sqrt(x * x + y * y), sqrt(x * x + y * y)

The advantage of describing an image like this is that it is independent of resolution - assuming that you describe the coordinate space in an abstract unit (such as “inch” or “width of a sheet of paper”) you can produce the image at both 72 dpi and 300 dpi and it will look good in either representation. Furthermore, you can also zoom in (by scaling the X and Y values) and the image will still look good (even if you zoom in thousands of times).

However, trying to create an image by describing it in mathematical terms like this is cumbersome to say the least. Suppose you had instead:

sqrt(x * x + y * y), sin(x), sin(x * y)

Few people would be able to look at that and immediately know that it would look like this:



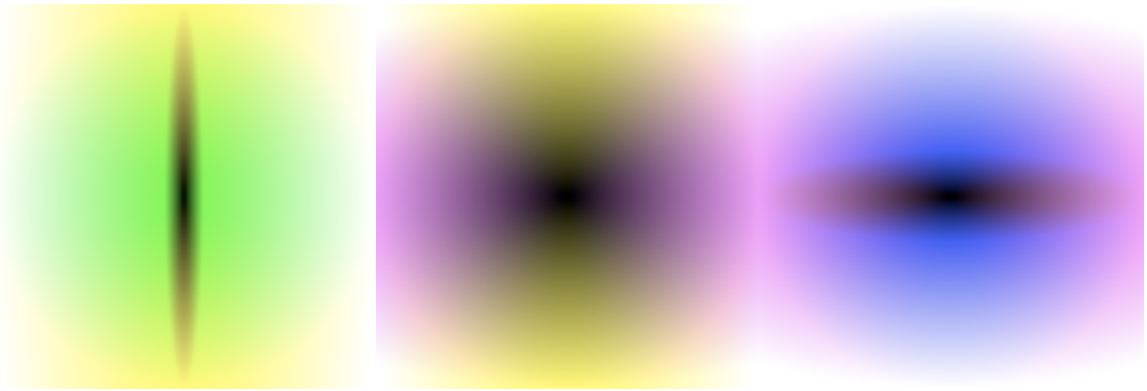
(or be able to do the reverse - take that image and figure out the underlying formula).

So quadrium provides a number of tools to create such images. In general, you would start with a sample image that provides some basic quality that you would like. In the above examples, the first image may be that “dark circular graident”. If you change a couple of small things in the formula, you get a different image. Sometimes these changes are to the underlying structure of the formula (like the way we changed “sqrt(x * x + y * y)” to “sin(x)” and “sin(x * y)”). Other changes are simpler - for example, we could change our original formula to:

sqrt(x * x + y * y), sqrt(4 * x * x + y * y), sqrt(x * x + 4 * y * y)

which gives us:

By making changes to these scaling values, we can create a wide variety of different images:

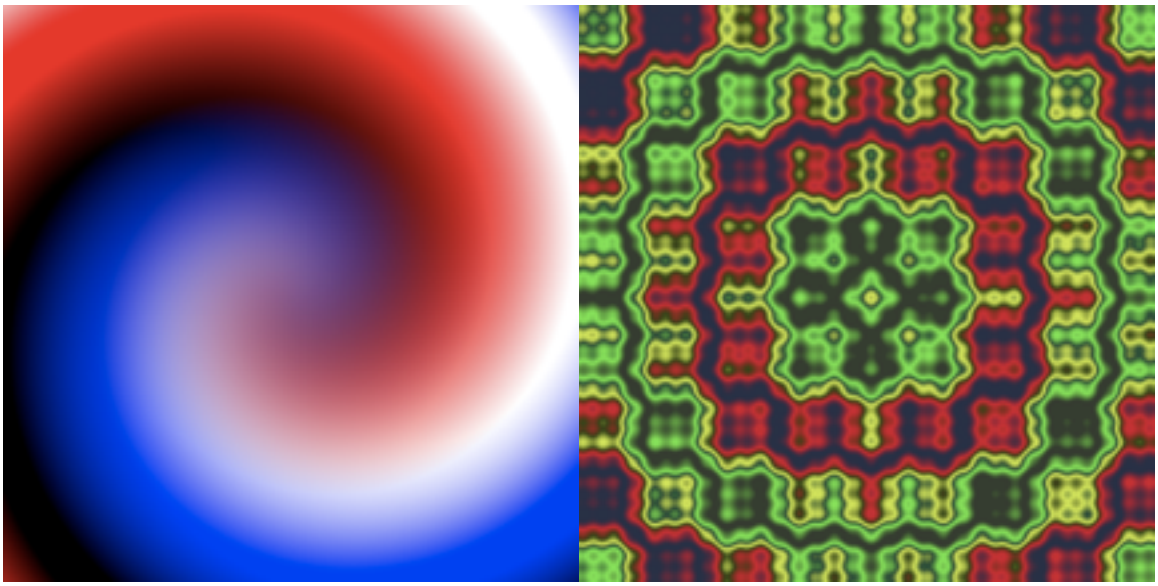


They are all similar, yet different variations on the same theme. quadrium takes this process and automates it for you - takes an image and can create a wide variety of variations, either by changing some of the basic parameters (such as the scaling factors we used in the last example) or by changing the underlying structure of the formula (like in the earlier example).

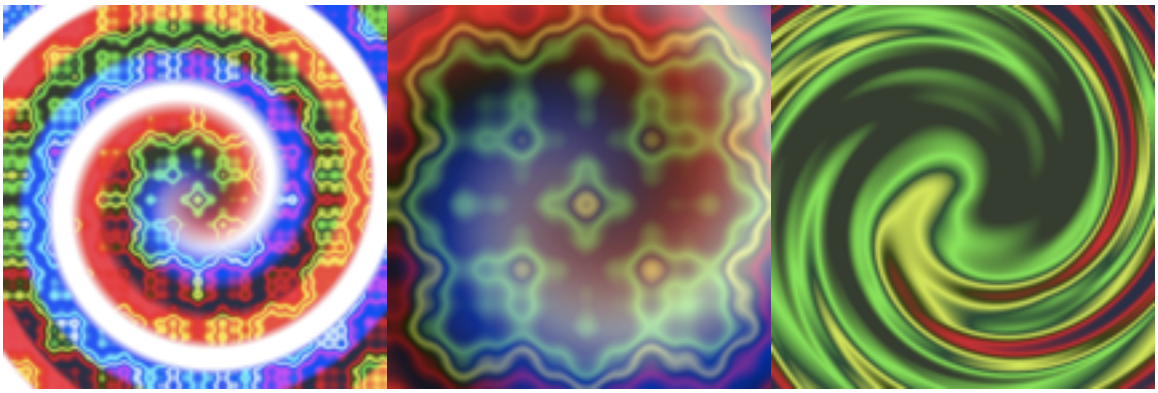
quadrium, however, is not limited to simple functions like we've used here - though it does include them, many of the functions are far more complicated, with much richer behavior (for example, a single "function" can represent a fractal, a woven pattern, or

mirrors that reflect the image through hyperbolic space that M.C. Escher would have been proud of). But changing one small part of the image formula into a different thing isn't all that quadrium can do - generating new images from a single starting image is powerful, but it's limited to just mutating the current image.

Suppose, however, that you were able to take two images and combine them together. For example, you could take the red component from image and the green and blue from a second - the result would show one image with a red hue and the other with a cyan hue. Or combining an image with blue swirls and green checkerboards could produce either something with blue checkerboards or green swirls (or swirls in a checkerboard pattern, etc...). All of these would be the result of "breeding" two images. For example, if we start with these two "parents":



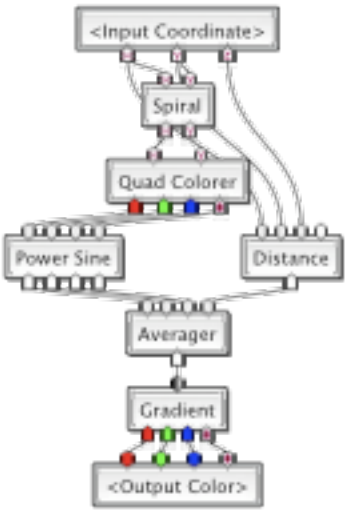
Any of the following "children" could result:



Obviously, the "traits" of the parent images come through in different degrees. By combining these two techniques, you are "evolving" the image, using both mutation as well as breeding (or what's know as "genetic algorithms" because them mimic the way

genes can be bred in organisms, allowing you to breed dogs that have extra floppy ears, or specific colors of roses).

Rather than representing the underlying formula as a long textual formula, quadrium models it as a series of nodes, wired together like a schematic. For example, that last child image is actually shown below:



In this example, the starting input coordinate (the location on the image) are fed down in to a “Spiral” and a “Distance” node, the “Spiral” goes on to a “Quad Colorer” which feeds a “Power Sine”, and something called “Averager” combines them all together, and then it goes to “Gradient” which finally connects to the “<Output Color>”. You can think of this diagram as the “DNA” of the image. If one were to write this out, it might look something like this:

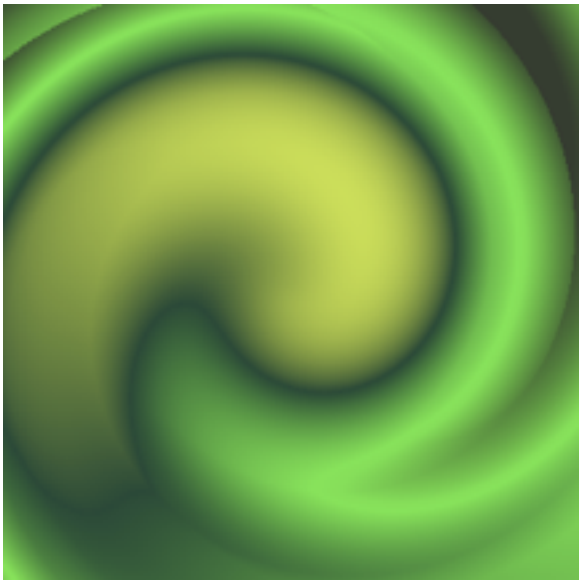
GradientRed(Averager(PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), Distance(x,y,z))),
GradientGreen(Averager(PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), Distance(x,y,z))), GradientBlue(Averager(PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), Distance(x,y,z))),
GradientAlpha(Averager(PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), PowerSine(QuadColorer(Spiral(x,y), Spiral(x,y))), Distance(x,y,z))).

Needless to say, the graphic representation is a far simpler model. However, you don’t need to actually work at this level of construction detail - many images can be created using just the simpler tools, but this level is available as you as you get more comfortable with quadrium. To work with this model, there is a special floating window that con-

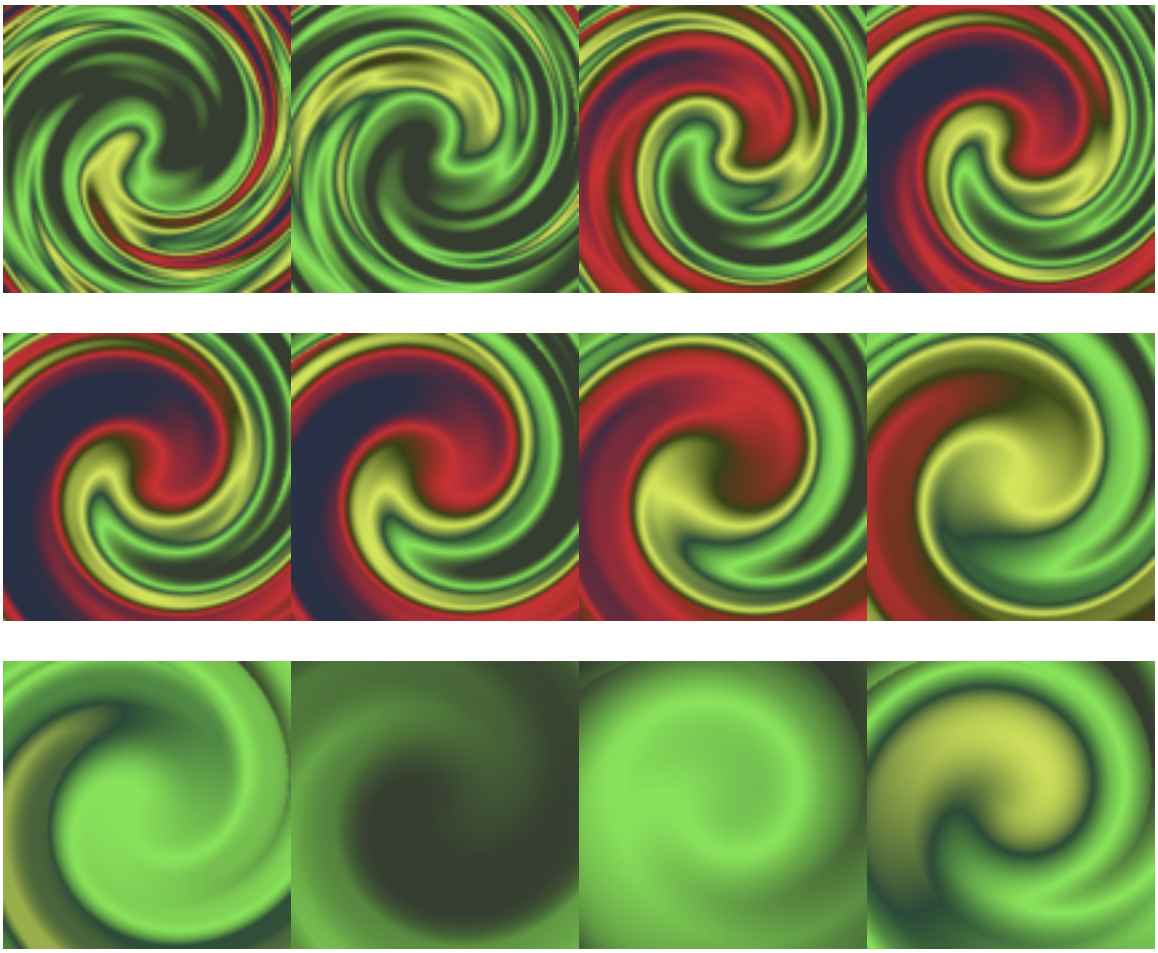
tains a large list of these nodes (over 230 of them), grouped by function. You can then just drag the nodes from this list, and drop them onto the structure, and then drag wires out from one nub on the bottom of one node, to the nub at the top of another, and quadrium will instantly update your changes.

Each of those nodes can represent a simple formula (like “Distance” which is “sqrt(x * x + y * y + z * z)”) but others are more complicated (the “Spiral” take the x and y coordinates, and than “spins” them by converting them to a distance from the center and an angle, and then adding that distance to the angle and converting that back to x and y coordinates - and math to model how to convert a single value to a gradient that goes between red, yellow and green (with some black between them) would take the better part of a page to write out. Many of those nodes include extra settings - like the scaling value we showed earlier, the “Spiral” node includes parameters to control the “tightness” of the spiral, how it rotates it, etc... (and of course, as you move the sliders representing these values, quadrium will provide live feedback showing you how this changes the image). Nodes such as the “Gradient” node also include color information (quadrium provides a rich gradient editor to help you create these gradients). Again, you can let quadrium randomly change these settings and present you with variations, or you can explicitly go in and to it yourself (for example, if you want to have blue in the image instead of green, the chance that quadrium will randomly change green into blue is pretty small, so you are much better off by just editing the gradient to change the green point to a blue point).

Since the difference between two images is often the difference between two sets of parameter settings, quadrium provides tools to take advantage of this. For example, one variation of the the spiral we saw is this:



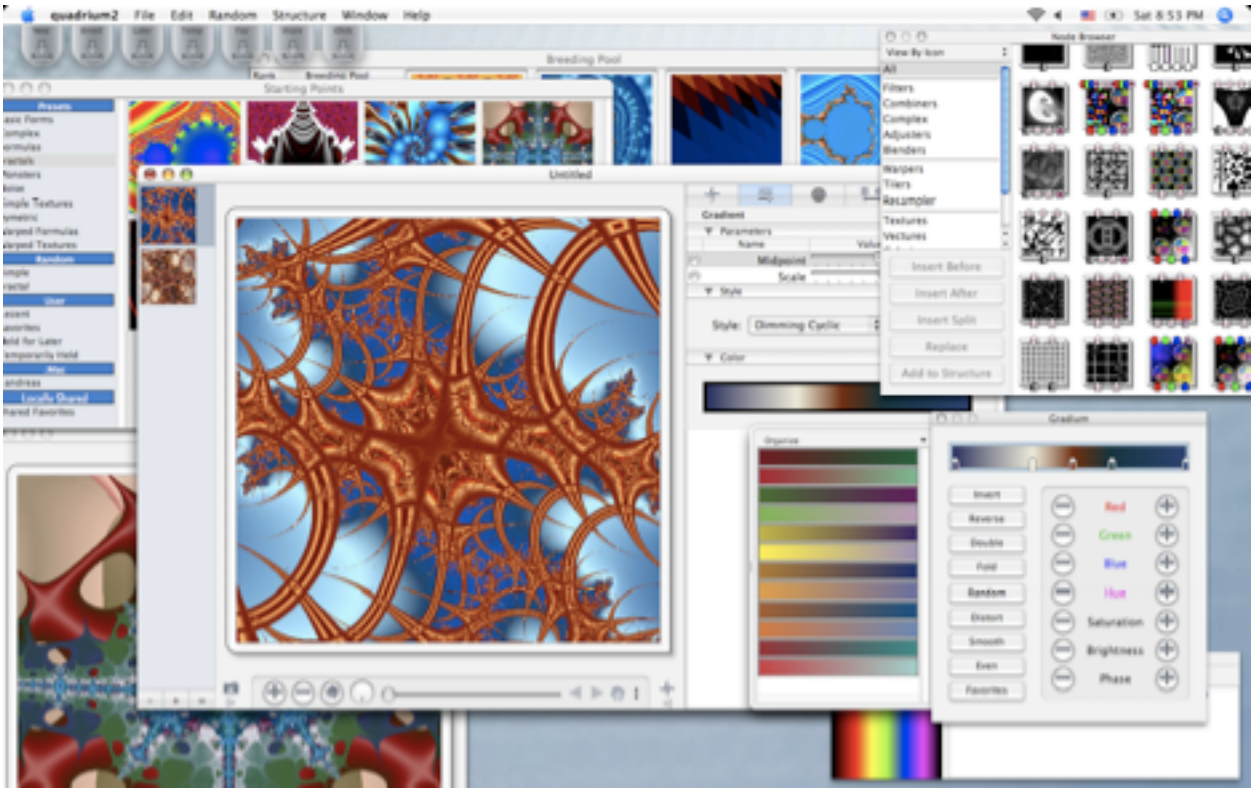
Since the only difference is a whole bunch of the parameters, we can find an image that is halfway “between” this and the original spiral by just changing those values only halfway. Or we can get 10% by only changing them a tenth of the way. This allows us to generate a whole series of images in-between these two images:



This allows us to do two things - first, is that we can find additional images with more control than breeding would allow (though this only works well when the underlying images have similar structures). The second is that we can use this technique to create a whole bunch of images, and combine those images together to create a movie, animating our work.

quadrium Window

There are a wide variety of windows that you'll see while using quadrium:



These include:

A "starting points" browser (used to select what sort of image to start from),

quadrium Image document windows,

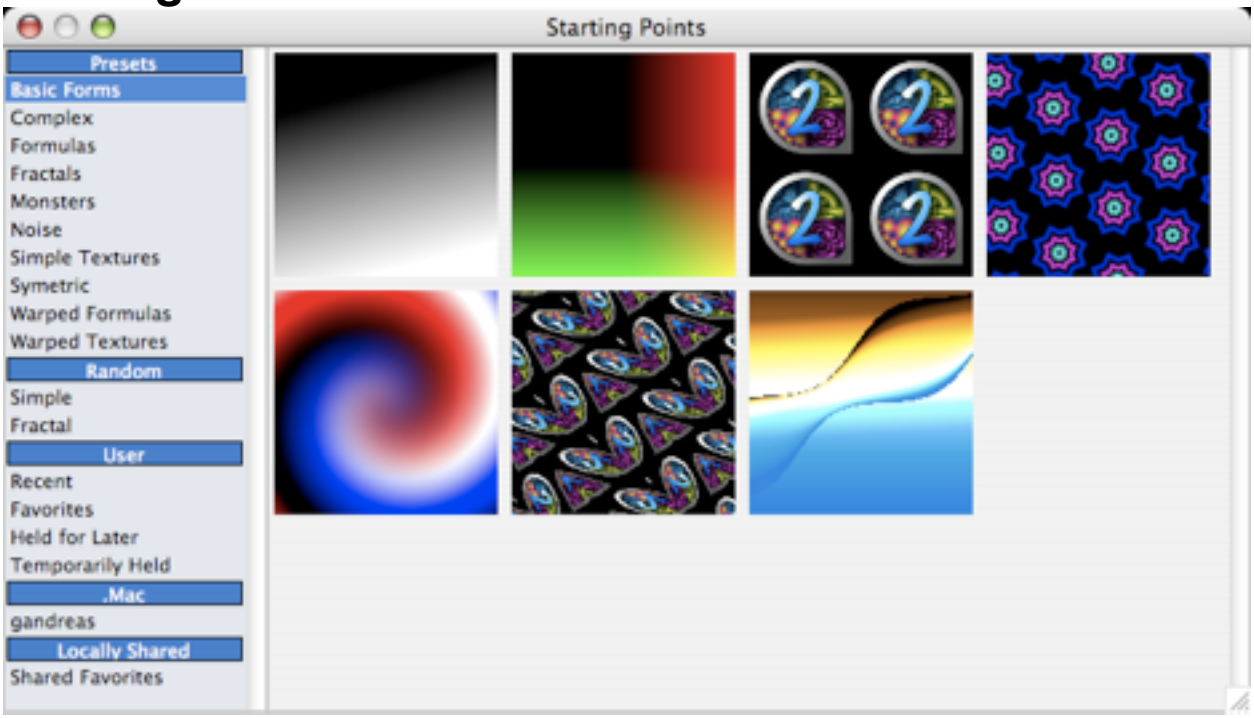
A "Breeding Pool" (used to combine images to synthesize new images)

A variety of floating windows (including a gradient editor).

A series of "drop targets" (located at the top left, below the menu bar) used to save temporary images, add them to the breeding pool, share the images, and a number of other actions.

quadrium makes extensive use of drag & drop - rather than a big series of nearly identical options in a popup menu or multi step "select the item, find the right menu and select the appropriate action from the menu", you simply drag the image from just about anywhere, and drop them either on one of the drop targets or in a number of other locations (such as adding a key frame to a movie or saving a snapshot of the current image, allowing you to easily switch between different variants).

Starting Points Browser



When you launch quadrium (or select **File > New from Starting Point...**), the *Starting Points Browser* will be shown. On the left is a list of various categories of images, and on the right are the images in that category. Clicking on these images will create a new document, using that image as a starting template (which you can then modify to create your own image).

The categories include a list of preset images provided with quadrium, broken down into groups. Below that are a several random categories - selecting one of them will generate a series of completely random images. You never know what sort of thing will be randomly created - many aren't all that interesting, but every once in a while a real gem will pop up. Both of these also provide a nice group of images that can be added to the "Breeding Pool" to add various traits to your images.

Next is a group of images that you've created. These include images that you've recently saved (note that if you want to open a recent document, you should use the **File > Open Recent** sub-menu - selecting the image from the starting points browser will create a new image based on that file), as well as your favorite images (as you work with quadrium, you will start to discover certain basic images that can easily be modified to produce interesting images). To add a new image to these favorites, drag the image and drop it on the drop target labeled "Fav".

There are then two similar categories: "Held for Later" and "Temporarily Held". As you create images you will often discover images that have potential, but aren't what you are looking for right now (or you otherwise want to get back to at a later date). If you drag it to the "Later" drop target at the top of the screen, it will add it to the "Held for

Later" category. This is similar to saving the file to disk and then reopening that file, but without cluttering up the desktop (plus it's a lot easier to tell from the thumbnail image presented in the browser which images are which). Plus, when you select the image, it will prompt you to remove that from the list (since now it is the "later" that you've held it for). You can also remove an image from this category by dragging it to the trash.

If, on the other hand, you are working with an image that might be useful (say you find two different variations in the Breeding Pool) you can drag an image to the "Temp" drop target. This adds it to "Temporarily Held" category. Just like the "Held for Later" category, you can select the image from the starting point browser, and it will also prompt for removing it from the list. Unlike "Held for Later", these images will go away when you quit quadrium (quadrium will prompt you to review these images if there are any in the list when you quit). This is analogous to taking a document in other applications, selecting all, copy, create a new document, paste, and then minimize the window, but with far fewer steps and less "dock clutter" (plus if you have more than a few such minimized windows, it doesn't take long before they become too small to tell which is which). You can also remove an image from this category by dragging it to the trash.

Finally are two other categories, use for sharing your images with others. The first is via .Mac's iDisk feature. quadrium automatically includes the iDisk for gandreas software - selecting that will fetch a series of additional sample images. If you have a .Mac account, your account will also be listed here, and any images that you drag to the "iDisk" drop target will be added to this list. To remove an image from this category, simply drag it to the trash. To add/remove other people's iDisks to this list, see the ".Mac" section of the preferences.

The other group is for sharing images locally, allowing you to show off your creations with your family or co-workers. Any other machines on your local network running quadrium that have sharing enabled will show up in this group as well. To allow others to see your machine, select the "Local Sharing" section of the preferences. To share an image locally, drag it to the "Share" drop target. Note that quadrium must be running to be able to share your images locally.

Besides using the starting points browser to create new images, the images in the starting points browser also work well for a source of images to add to your breeding pool, or be used in a document's tweener (used to blend together two images).

Drop Targets

Below the menu bar are a series of translucent “Drop Targets” which are used for a number of purposes. In general, dragging an image and dropping it on an appropriate tab will add it to one of the starting points browser categories, share it, or set it aside for later. From the left, the targets are:

New - dropping an image here will create a new document with the contents of this image. For example, to create a new image from something in the breeding pool, drag it to this drop tab. Double clicking this target will open the starting points browser.

Breed - dropping an image here will add it to the breeding pool. Double clicking it will show the breeding pool window.

Later - holds the image for later (this saves it to disk so you can come back to it at a later date). These images are available from the “Held for Later” category of the starting points browser, which is selected if you double click the target.

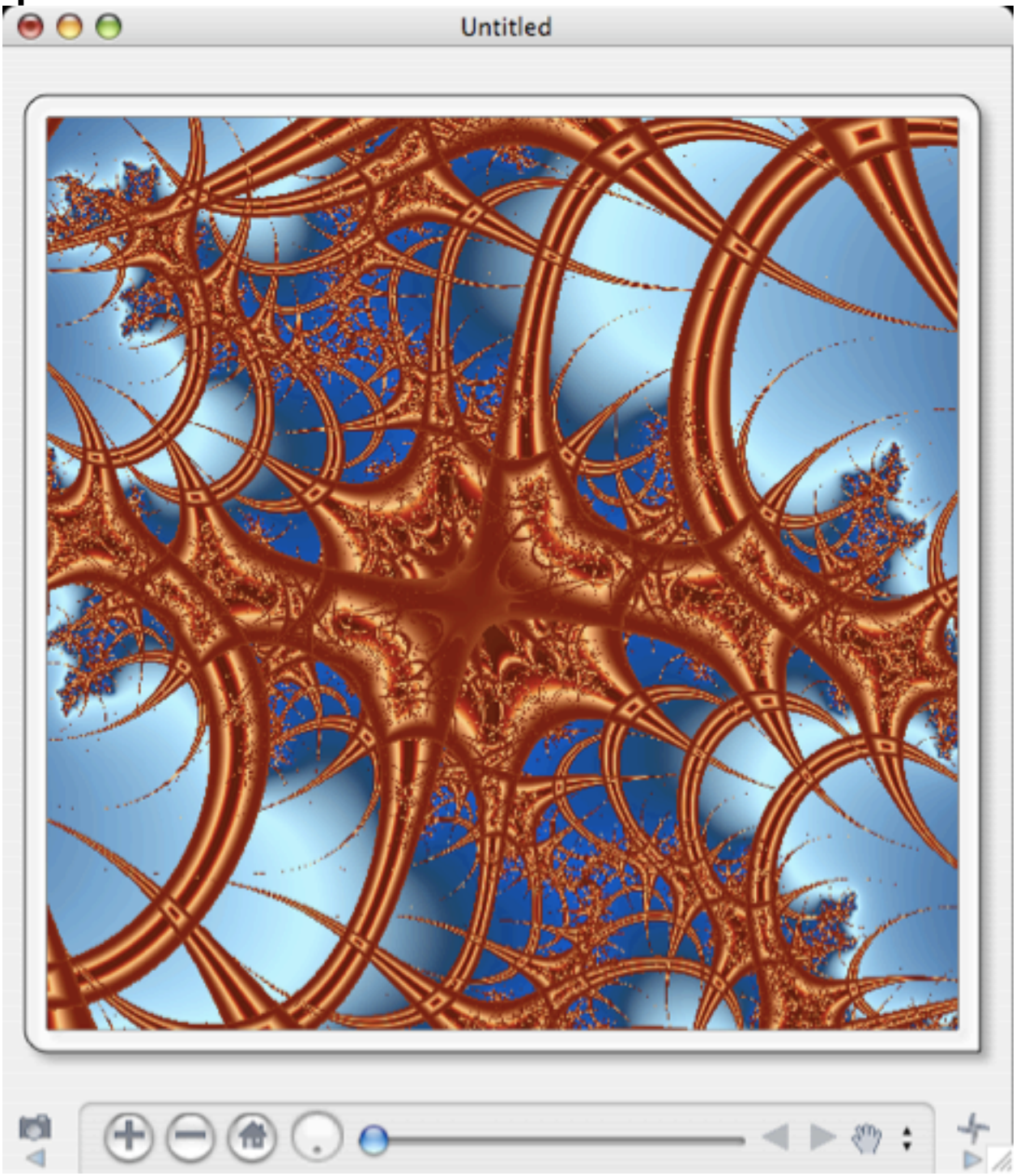
Temp - temporarily stashes the image away, similar to creating a new document with the image and then minimizing the window (but without the clutter). These images are available from the “Temporarily Held” category of the starting points browser, which is selected if you double click this target.

Fav - adds the image to the “Favorites” category of the starting points browser, which is selected if you double click this target.

Share - adds the image to the group of images that are shared on you local network with family or co-workers who are also running quadrium. Double clicking this target will open the starting points browser with the “Shared Favorites” category selected.

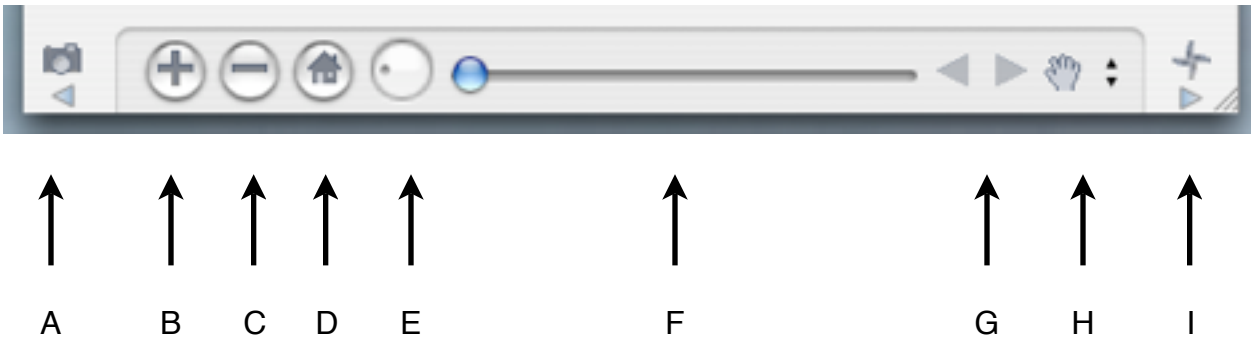
iDisk - writes the image to your iDisk, allowing others to access them (requires a .Mac account to be able to share images). Double clicking this target will open the starting points browser with the .Mac category corresponding to your account selected.

quadrium Documents



When you first create an image or open an existing document, the document will be shown as above. The majority of the window is taken up by the image itself, bordered by a frame (it is by clicking and dragging this frame that you can drag the image to drop target, since clicking in the image is used to perform actions such as panning the im-

age). Along the bottom are two “expanders” in the corner, and a series of controls. These are as follows:



- A - Snapshot Expander. Click to show/hide the snapshot panel. Images can also be “dragged through” this, showing the snapshot panel.
- B - Zoom out from the image. Holding down the button will repeatedly zoom out. Holding the option key while clicking this button will zoom out in finer steps.
- C - Zoom in. Same as “Zoom out” but zooms in.
- D - Home. Clicking this will return the view point to the default location with the default level of zooming and no rotation.
- E - Rotate - turning this dial with rotate your image. If you control-click on this button, a menu of common rotations are available (in 45° increments).
- F - “Z Coordinate”. Sliding this slider will change the “Z” value of the image (not many images take advantage of this feature, but it provides an easy way to have an input value that can be readily changed).
- G - Point of view history controls. Clicking the forward/backward arrows will move you forward/backward through a list of various views that the image has been in. For example, if you move the image and decide that you didn’t mean to, you can click the left arrow to go back as step. Note that because these view movements don’t actually change the underlying image, they aren’t recorded via the undo/redo mechanism. Instead, these controls can undo/redo those actions.
- H - Current View mode selector. There are five different modes (you can temporarily switch between modes by holding down the appropriate key, or holding down the shift key with it will change it):
  -  - Pan (Space). Dragging in the view will move the image. Holding down the shift key while dragging will constrain movement to either horizontally or vertically. Holding down the option key will move “fine scale”.

-  - Select Area (”a”) - Dragging in the view will specify an area of the view to zoom into. Holding down the shift key will select from the center.

-  - Zoom (”z”) - Dragging in the view up or down will zoom in or out.

-  - Rotator (”r”) - Dragging in the view will rotate the image about the center. Holding down the shift key will restrict rotation to 45° increments.

-  - Magnifier (”m”) - Clicking on the view will show the built in view magnifier. While the mouse is down (showing the magnifier), you can use the up/down arrow keys (or scroll wheel on your mouse) to increase/decrease the magnfication.

-  - Eye dropper (”e”) - Clicking in the view will show the color at the pixel underneath the tip of the eye dropper. If a color well is currently selected, this will set the color (possibly redrawing the image). Similarly, if a parameter that is selected is part of a complex value (i.e., has a blue-green oval background in the parameter list), it will set both components to the current location that the eye dropper is. If you hold down the shift key while doing this, it just offsets the coordinate relatively (and the option key will cause this to be a “fine” movement). For example, if you have a Julia Fractal node selected and are editing it’s parameters and select either of the “Seed” parameters (”Seed Real” or “Seed Imag”), clicking the eye dropper will change this seed value, allowing you “drag” the fractal through the the “seed space”.

- I - Details expander. This will show/hide the panel with all the various image details, including the underlying structure, node parameters, mutations, tweener, movie, and general information tabs.

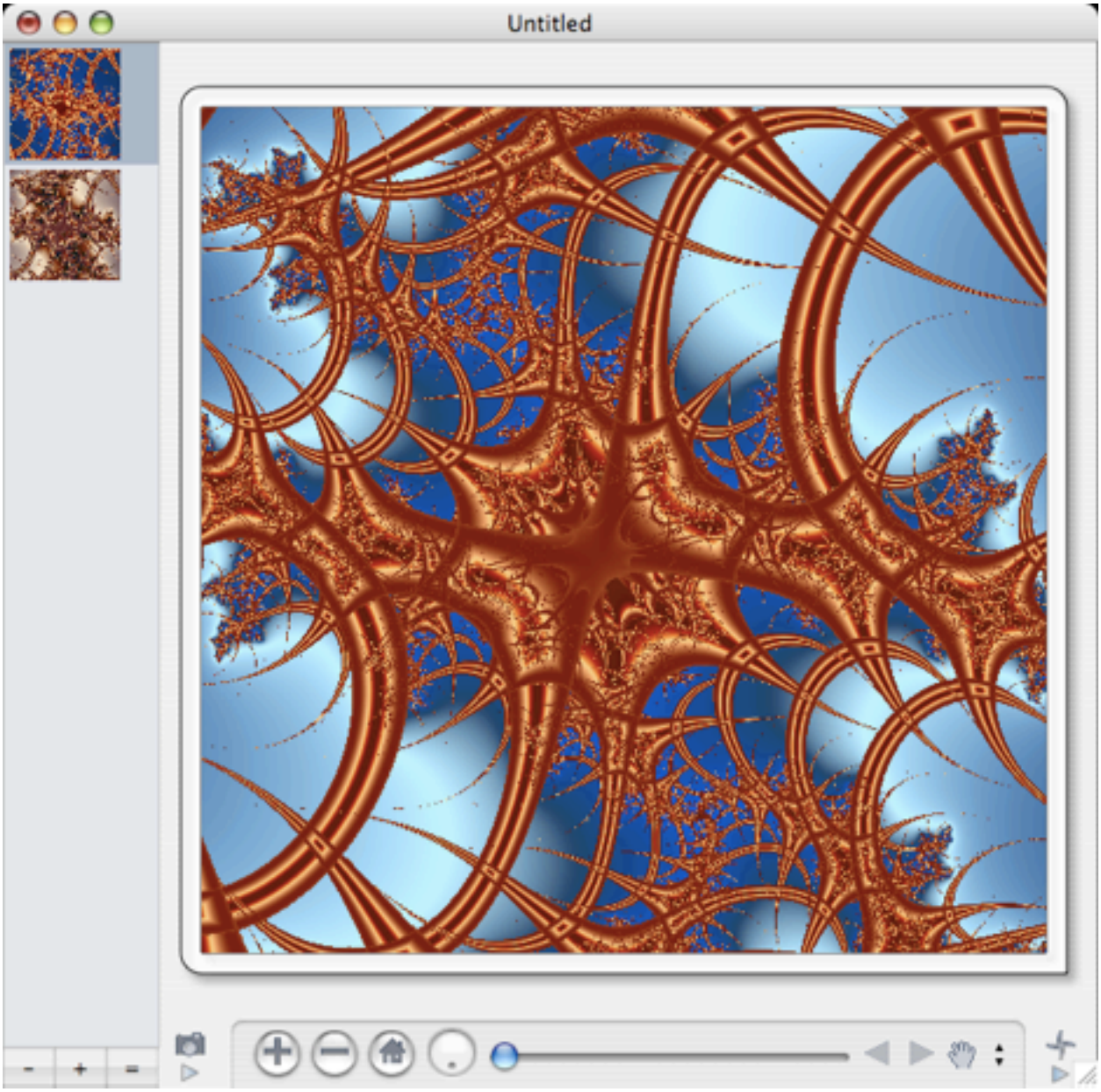
Rulers

If you hold down the command key while clicking in the main view, rulers will appear along the outer edge of the image. This holds true for all the various possible view modes. The rulers are in whatever unit the document is in (as specified by the document information tab of the details panel).

The rulers are designed to allow you to align the image such that various features can be located at specific areas on the final rendered or printed image.

Snapshot Panel

Clicking on the snapshot expander will show/hide the snapshot panel:

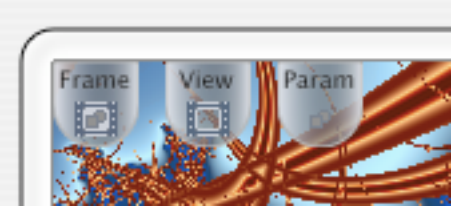


The bulk of this panel is composed of the snapshot list. This is useful to either various versions of your image as you create it (making it easier to go back to some good state after having taking the image in the wrong direction) or to save several different variants (so if you find that the image in blue looks good, you can save a snapshot of the blue version and one of the green version).

To take a snapshot of the current image, you can click the “+” button in the snapshot drawer (or take advantage of the menu item **File > Take Snapshot** or its shortcut “command-option-s” or the even easier “+” key). To remove a snapshot, select it and hit the “-” sign (or “-” key). To restore the current image from a snapshot, double click on it, or click on it and hit the “=” sign (or “=” key). You can also drag an image from the snap

shot list and drop it in the main image view (or drop it anywhere else, such as a variety of drop targets, movie frames, tweener, etc...).

If you drag the image to drop it in the main view (be it from the snapshot list or anywhere else) three small drop targets will appear at the top left corner of the view:



These allow you dissect image you are dropping and only take a small portion of it:

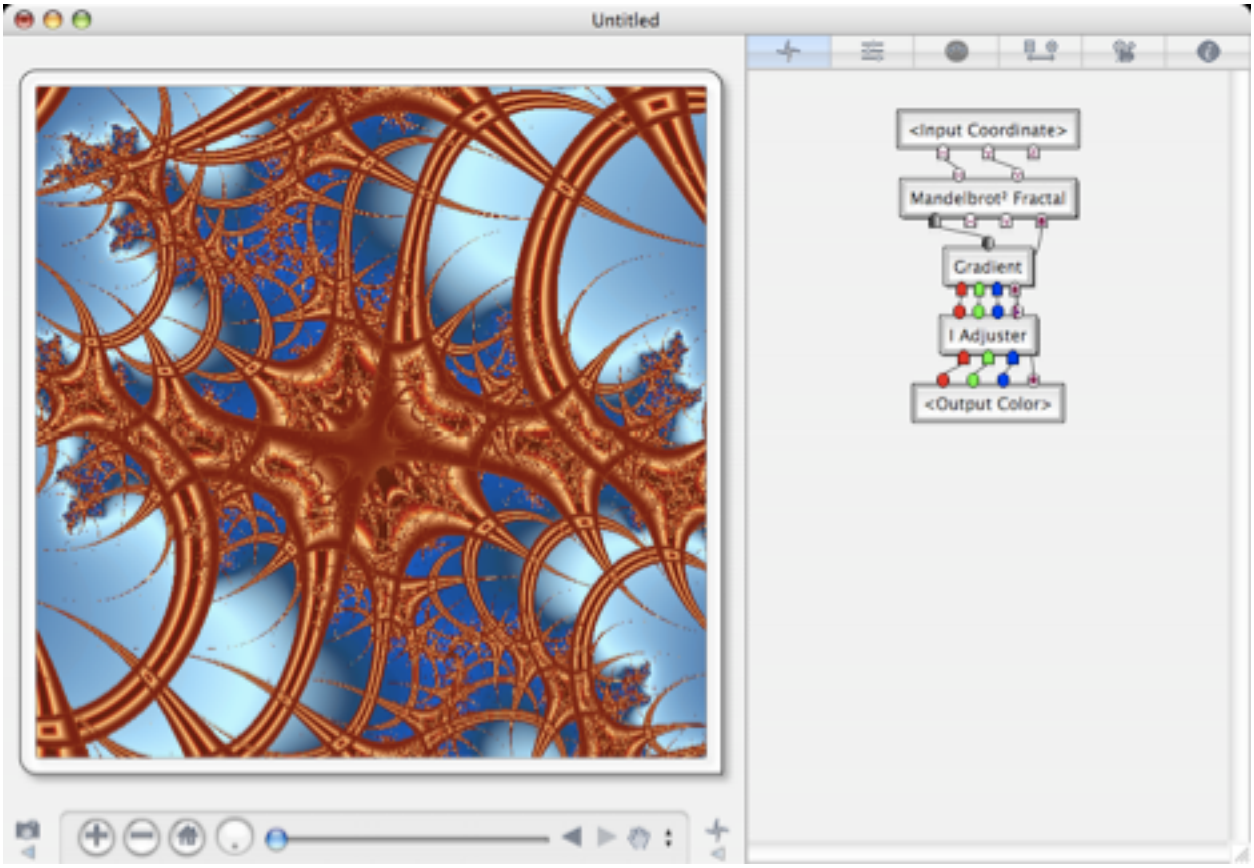
•**Frame** will take as much of the information from the image being dropped as possible, but retains the structure of the current image (so it will extract the view information such as coordinate, zoom, rotation, as well as various other information such as parameter values). Note that parameters that aren't found in both the dragging image and the target image are effectively ignored/unchanged.

- **View** just takes the point of view from the image being dropped (especially useful if you are working with certain fractals where if you find an interesting location in one fractal you can use this to focus the second fractal on the same location).

- **Param** will take the parameters (where possible) from the image being dropped and apply them to the target image. The view and structure will be unchanged.

Dropping an image anywhere else within the main view will replace it completely (you can, of course, undo this action).

Details Panel
Clicking on the details expander will show the details panel. This panel is composed of a series of tabs, as shown below:



This panel is composed of six different tabs, which can be switched by clicking on the icons at the top (you can also hover the mouse over these tabs while dragging an image to switch tabs as well). In order, these tabs are as follows:

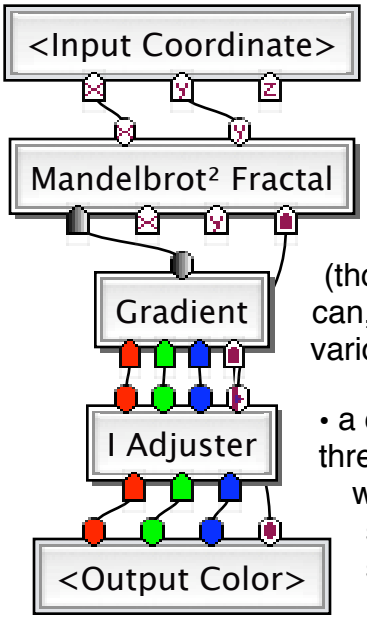
- **Structure** - shows the structure of the image's formula, with nodes and connections between them
- **Parameters** - shows the various parameters of the selected node (including both the sliders and other settings such as colors and various options)
- **Mutations** - generate various mutations of the current image
- **Tweener** - generate images "in-between" this image and other images.
- **Movies** - create a movie from a series of variations on the current image.
- **Document Information** - set thing such as the document size, resolution, and author and copyright.

You can also switch between these tabs by typing the keys 1-6, and double clicking on a node in the structure tab will automatically switch to the parameters tab.

Working with Structure - Nodes and Wiring

When you select the first tab, it shows you the underlying structure of formulas used to generate the image. This is represented as a series of nodes, with small “nubs” wired together, much like a circuit diagram:

In this example, the input coordinate is fed into a fractal node, one of whose outputs is fed into a gradient, the other is combined with the gradient output and used to adjust the “I” channel of the color. That output is then shown on screen. This flow always goes from top to bottom - there can be no loops (quadrium will not allow you to connect a wire in a way that would form such a loop).



You’ve probably noticed that the various input and output nubs are keyed with different colors, letters, or patterns. These provide a hint as to what can be found on that nub (thought these hints are just that - suggested uses, and you can, for example, connect a red output to a blue input). The various possible hints include:

- a coordinate (such as “X”, “Y”, or “Z” to indicate one of the three possible coordinate locations that are normally found, with “X” and “Y” corresponding to the horizontal and vertical screen coordinate of the image, and “Z” being tied to the slider)
- a color - red, green, blue, for those respective color components.
- A grayscale value - these values are normally in the range of 0 to 1 (though nothing requires this).
- Various special other nodes, such as an alpha component (purple inside the white frame) or “adjustment” value (the purple “notch”).

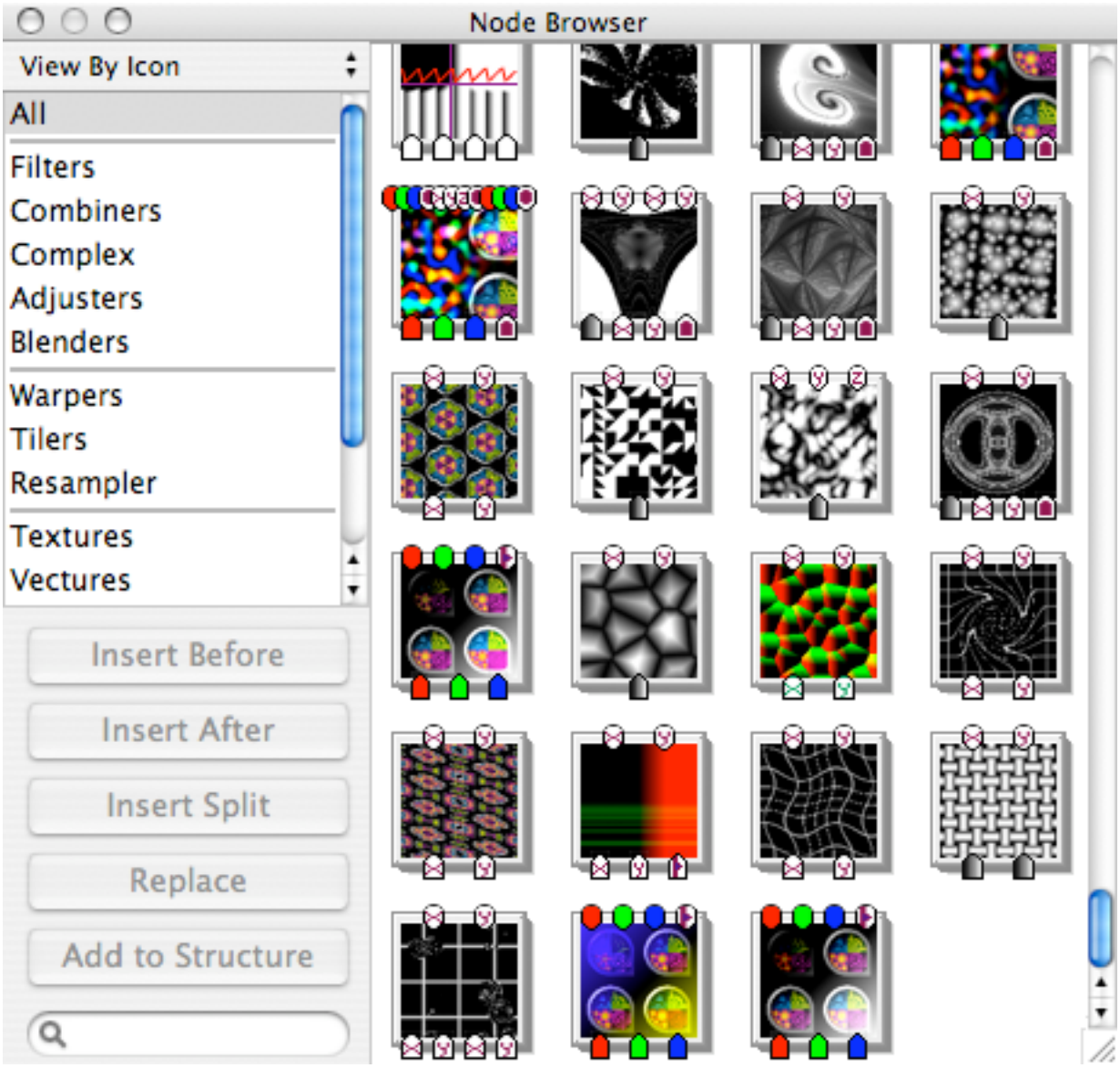
You can alter the appearance of this diagram from the **Structure > View Structure** menu, selecting between text, small icon, and large icon.

To add a new wire connecting two nodes, click on the output nub (lower side) of the node and drag to the input nub (top side) of the destination node. As noted above, you can’t form a loop. To remove a connection, click on the input nub and drag away the connection that goes into it, either connecting it to a different input, or just simplely discontecting it all together by releasing the mouse when it isn’t pointing to anything. You can have multiple connections going *out* of a single output nub, but only one going *in* to an input nub. If you hold down the shift key while clicking on the ouput nub, it will attempt to connect all the outputs of node to all the input nodes that you drag to, allowing to rapidly connect a series of wires up in parallel. If there is only a single output nub in the node, it will attempt to connect *all* of the inputs to that output. If there are more nubs

in the output or input, the extra are ignore (except in the case where there was only a single output nub).

Double clicking on a node will switch to the parameter editing tab (see below). You can also delete that node (hitting the delete key or menu item), or, by holding down the control key (or right clicking) while clicking on a selected node, you can get a contextual menu with a variety of node options (including being able to copy just that node to the clipboard, because otherwise “copy” will attempt to copy the image).

To add a new node into this structure, you need to use the **Node Browser** floating panel. It appears by default when you click in this pane, or you can explicitly show it using the **Window > Node Browser** menu item. The node browser looks like this:



On the right is a large list of the available node. The view shown is the “icon” view, showing the larger icon version of the node (making it easier to rapidly identify a wide variety of nodes). The menu at the top left can be used to switch between icon, small icon, or text modes, helping you pick the layout that maximizes the nodes shown based on the size of the window.

On the left is first a series of node categories - all nodes fall in one or more of these categories. Selecting one of these categories will filter what nodes are shown, making it easier to find specific nodes (since there are some 230+ nodes available). These categories include:

- All - shows all of them
- Filters - shows nodes that take four inputs, and perform the same operation independently to each of them in parallel (for example, take the sine of the input as the output).
- Combiners - Nodes that take four inputs and combine them down to a single input. These are useful as way to combine grayscale images, for example.
- Complex - Nodes that perform complex arithmetic. Usually they take four inputs (two sets of complex numbers, with both real and imaginary components) and have a pair of outputs (a single complex number, again with both real and imaginary components).
- Adjusters - Nodes that adjust RGB colors, taking an RGB color input and an additional “adjustment parameter”, and produce an RGB color output. For example, this can be used to lighten, darken, or adjust the hue of an image
- Blenders - Combines two sets of RGBA colors, producing an output RGBA color. Some take extra parameters to control how the colors are combined, for example, to select one input set or the other.
- Warpers - Takes a coordinate (either X and Y, or X, Y and Z) and produces a second coordinate. The results often twists, distorts, or otherwise warps the image. Usually, however, all points of the image “upstream” are visible “downstream”
- Tilers - Similar to Warpers, but the output is usually sliced and diced and then repeated. Since tilers repeat the image, parts of the image “upstream” may no longer be seen “downstream”.
- Resamplers - Change the underlying resolution of image, similar to warpers, but often with the discrete edges that are found in tilers.
- Textures - A rich set of nodes that takes a coordinate (either XY or XYZ) and produce a single grayscale output (though sometimes additional related output). These produce a wide variety of patterns and other textures.
- Vectures - Similar to textures, but the output, instead of being a grayscale value, is a two-dimensional XY value, which can be used to manipulate an image. Where as tex-

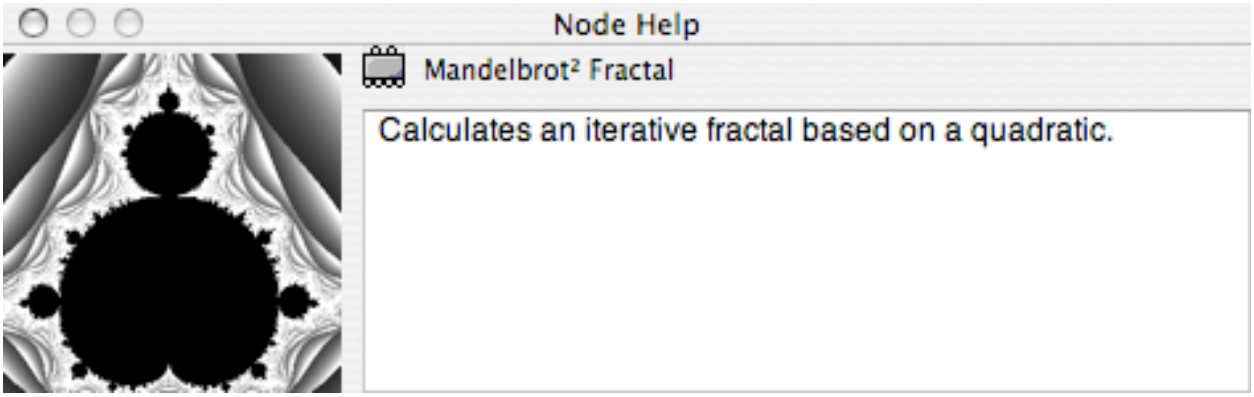
tures are effectively flat and colors, these are sort of multi-dimensional and create “spatial” texture (such as making it appear that rain drops are on the image).

- Colorizers - Take a grayscale value (or other non-color component input) and produce a color (RGB or RGBA) output. A simple example is the gradient node that converts grayscale to color.

- Fractals - A collection of nodes that produce fractal images, ranging from simple Mandelbrot images, to more complex fractal images (including iterative escape fractals with user provided formulas). See the *Reference Manual* for more details about fractal nodes, especially how iterative escape fractals work.

- Noise - A variety of nodes that work on various random noise functions, which can add static or other randomness to the image.

Note that a given node can fall into more than one category. At the bottom is a search field that allows you to find a given node by name (so if you want to find a “hue adjuster” you can just type “hue” to quickly find it - note that this will filter based on the selected category). When you select a node in the node browser (or a node in the structure), a node help window will be shown with more information about that node, including a larger thumbnail of what sort of image it creates:



If this window is closed, it can be reshown at any time by selecting **Help > Show Node Help**. This information, as well as more details, is collected in the *Reference Manual*.

To add one of the node to your image, click an drag from the node browser and drop in the structure view. Where you drop it determines how it is hooked into your image:

- Dropping it in a blank area will add it to the structure, but not have it connected to anything.

- Dropping it on the top half of a node will insert it “before” that node - all inputs that come into that destination node will be wired into the input of the new node, and the output of the new node will be wired to the input of the new node.

- Dropping it on the bottom half of a node will insert it “after” that node - all of the outputs from that destination node will come instead from the outputs of the new node, and the inputs of the new node will be connected to the outputs of the destination node.

- If you drop it on a node itself, it will completely replace that destination node (deleting the old destination node). If you hold down the command key, this will “merge” in the node - all the settings that exist in common between the original and new node will be copied (for example, allowing you to easily change the type of a fractal without having to redo all the settings for the traps and coloring).

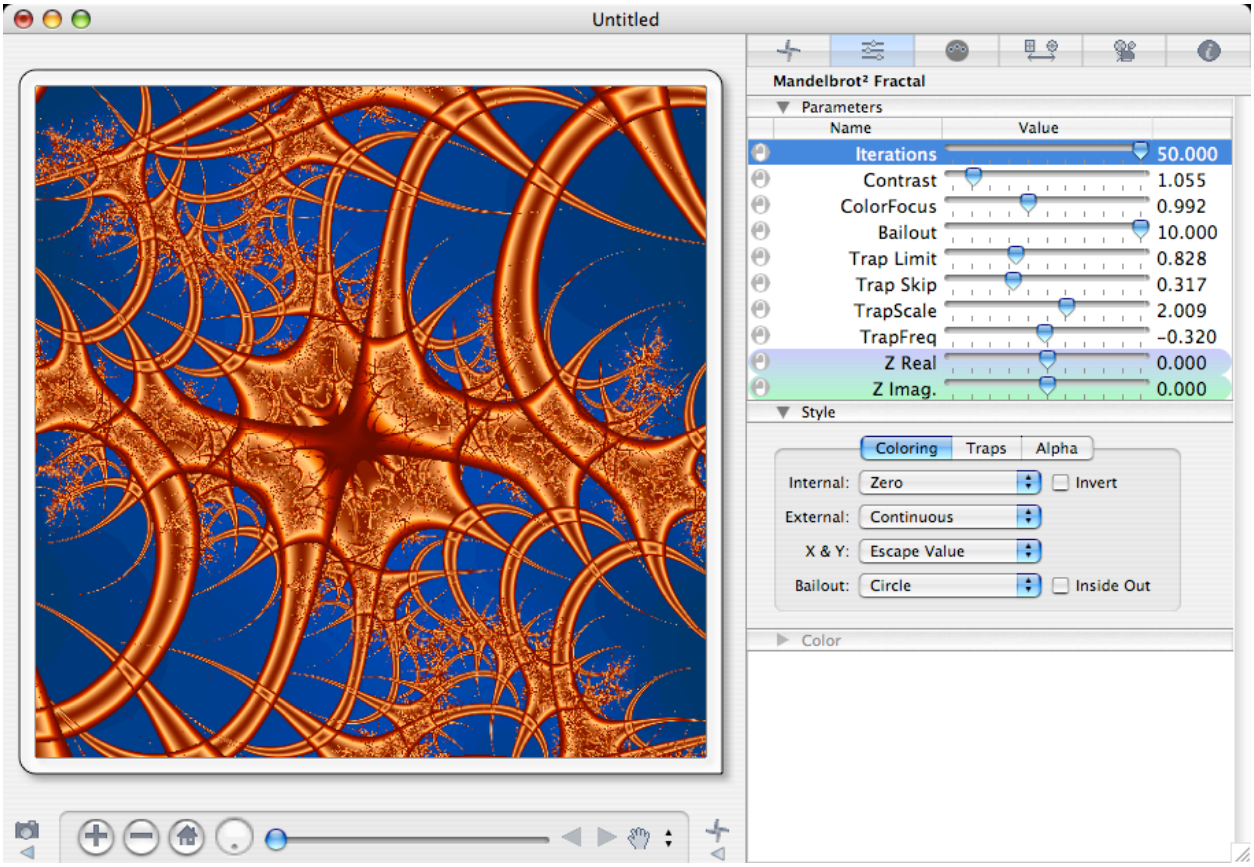
- If you drop it on the left side of a node, it will “split” the structure at that point - all of the inputs of the destination node will also be connected to the inputs of the new node (effectively “splitting” each of those connections at their sources). The outputs of new node will not be connected to anything (since the outputs of the destination node will be unchanged).

- If you drop an entire snapshot (as dragged from the snapshot area, mutations, starting point, etc...) and drop it on the side edge of the input node, it will merge the two images together, alpha overlaying/underlaying the new image with the original (depending on if you drop it on the left or right side of the input node). If there is no useful alpha in the images, you’ll probably want to adjust the transparency setting of the blender node to allow both images to be seen.

Similar effects can also be done by selecting the destination node in the structure, and also in the node browser and clicking the respective buttons on the lower left.

Editing Node Parameters and Settings

The second tab of the details pane contains the parameters and settings for currently selected node. These will vary widely depending on what node is selected, but are broken down into three categories - parameters, style settings, and colors:



(This example, the Mandelbrot node, only contains parameters and style, which is why the color section is disabled).

The parameters are represented in a table - each parameter has a name, a slider (with associated numeric values) and a “lock status”. Moving those sliders will adjust that parameter (and associated numeric value), redrawing the image as you do so (live feedback). You can also double click on the numeric value and edit that directly as well (since it may be difficult, for example, to set a slider to exactly 1.53). By directly editing the numeric value, you can also expand the range of the slider for most of the parameters - so if the slider goes from -2.5 to +2.5 and you enter 5 in the numeric field, the slider will then range from -2.5 to 5 (some parameters, have specific hard coded maximum or minimum values, so if the value is limited back to that, you’ll know why).

To fine tune the slider, you can use the option key - click on the slider with the option key down and then **release the option key** and move the mouse - the slider will move at a fraction of the rate of the mouse.

If you notice that some parameters have shaded background (like “Z Real” and “Z Imag” as shown in the screenshot above), this indicates that those two values compose a complex number. If you have either of those values selected, and switch to “eyedropper” mode in the main view, clicking the eye dropper will “pick up” the complex coordinate that you are clicking on and set the pair of sliders to that value. As mentioned in

the “Eyedropper” section above, the shift key will offset that value (rather than set it), and the option key will “fine tune” the movement.

The lock icon on the left of each row is used to lock that parameter. If a parameter is locked, you can not accidently edit it (without unlocking it first). More importantly, a locked parameter is never randomized when mutations are created (see below for mutations). For example, if you decide that you like the “Z Seed” value in the Mandelbrot fractal, but want some mutations based on the other values (such as trap parameters) you can lock those two Z values down so they won’t be changed.

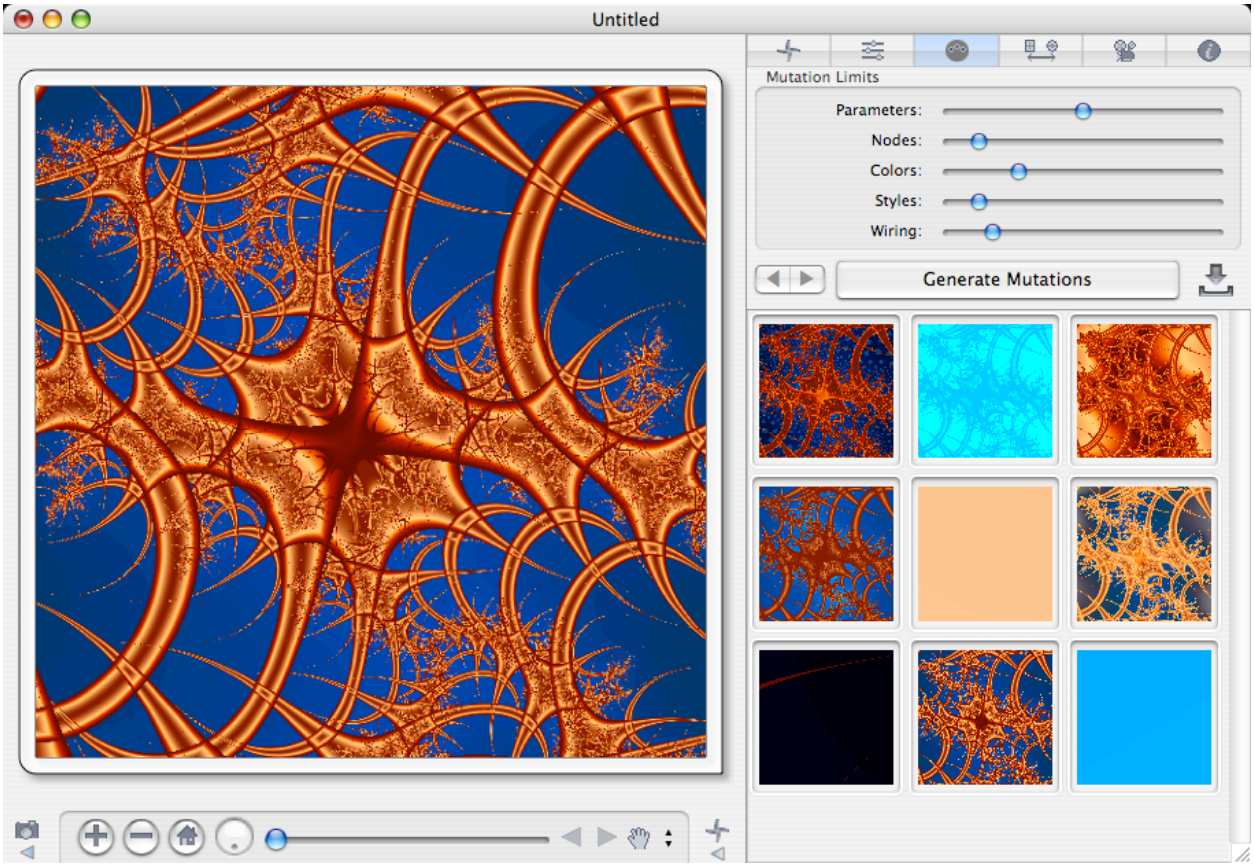
Note that there are a wide variety of parameters, some with meaningful names, other less so (often with technical means that isn’t completely clear). If you don’t understand exactly what a given slider is suppose to do, don’t worry about it - just drag it back and fourth and see how it changes the image. If you like what it does, great. If you don’t, just hit undo.

The “Style” section is used to set explicit style variations. Most nodes will only have a simple popup menu item or two - iterative fractal nodes have a boatload of settings to tweak (so many so that it is broken down into three different tabs). These designed to be discrete settings (unlike the continuous values that a parameter uses), and often has a textual description of what the setting is (a small handful allow the user to type in various string values). Switching between these value often has a radical effect on the image, and not the gradual changes that the parameters provide (this is important for tweening covered below, because these settings aren’t “tweened” because they are discrete values that have no “inbetween” value).

The “Color” section is used to indicate what colors or gradients are used by the node - either a single or series of color wells (clicking on them shows the standard color picker panel), or a gradient (clicking on it will show the gradient editor described below). Note that unlike settings, these values normally do allow tweening to take place (because halfway between two colors is a third color).

Mutations

The third tab in the details pane provides a way to create mutations of the current image:



The techniques involved in generating these mutation are detailed above in the "**How it Works**" section, but briefly, the current image is taken, and various parts of it are changed in different ways:

- Parameters values can be change, much like sliding the slider in the details panel.
- Nodes can be changed - a similar node can replace any given node in the structure. For example, if there is a "Sine" filter node, it could be replaced with a "Gaussian" filter node.
- Colors can be changed, either individual colors or gradients.
- Style values can be randomized.
- Wiring between nodes can be change - for example the red and blue connections can be swapped.

How likely each of these things are to happen is controlled by the sliders at the top of that pane. Sliding the slider to the left reduces the likelihood of that class of mutation of happening, and to the right increases it. At the bottom is a grid of nine possible mutations of the current image. Like other places, you can drag an image from that grid, dropping it any of the drop targets (or just drop in the main view to use it as current image), and you can control-option click on it to get a larger view of the image. If you

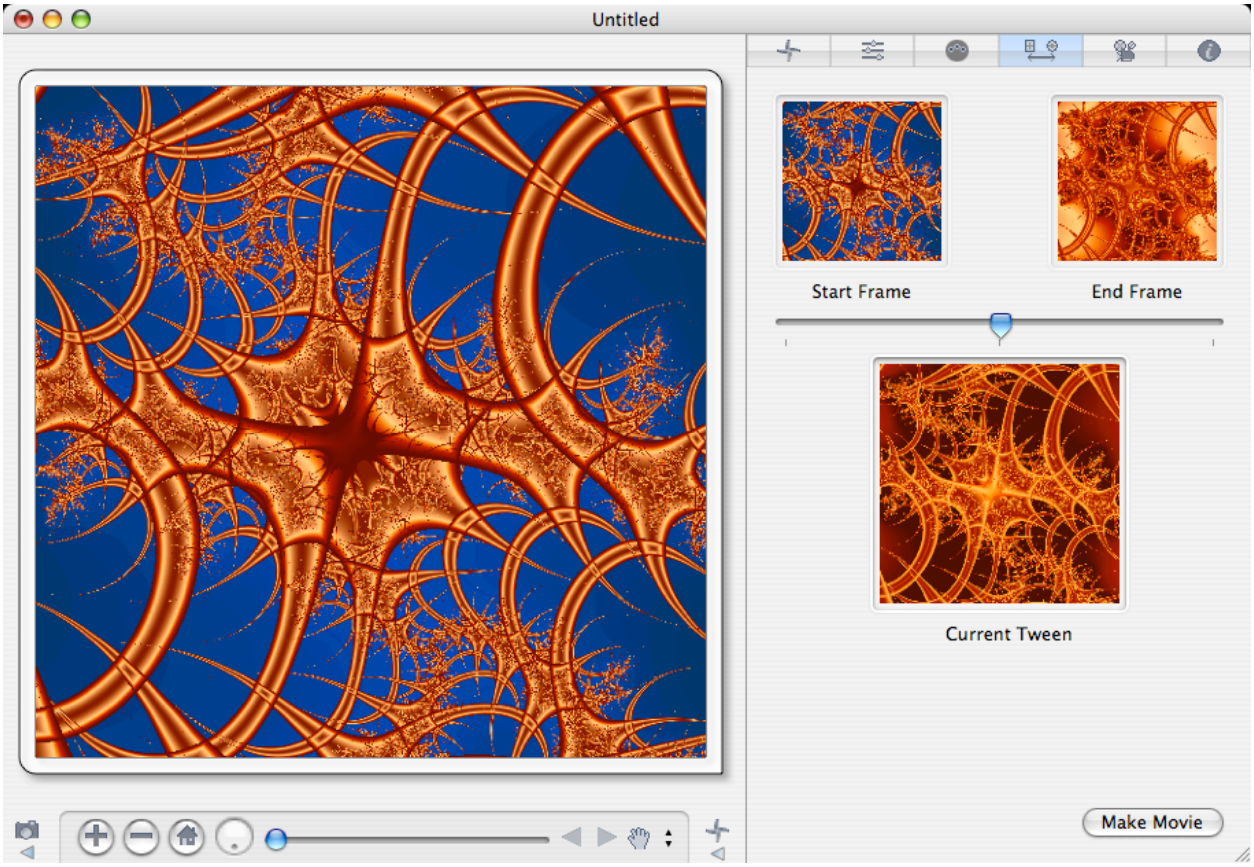
double click on that image, it will “burrow down” into the mutations, using that mutated image as the basis for generating future mutations. For example, if you double click on the top right image above, you’d get mutations based on the red and gold fractal image.

Between these two are a series of controls. The big button will generate a new set of mutations in case you don’t like any of them. As is often the case, if you trigger happy and click this one too many times and would like to “go back” to the previous set, the arrow buttons to the left of the this act as a history control. You can use this to go backwards and forwards through the history stack.

Finally, on the right is a drop target to use to make mutations from - the most common use of this is to drag a snapshot from snapshot pane and drop it there (you can also drag an image from the grid and drop it there, which is the same as double clicking it, but more work).

Besides providing new possible images, mutations generated here work well for tweening and frames of a movie (since they are based on the original image, there is a “low impedance” between them). Don’t forget that images can be “dragged through” the tabs at the top, making it easy to drag the mutation to a different tab to drop it (or dragged through the snapshot expander button to show the snapshot pane if hidden).

Tweening
Tweening is a process of taking two images and finding the image between them:



For example, we can use this to find an image that is halfway between the original blue background fractal with a dark center, and the “red and gold” version that we saw in mutations section. To do this, we drag the original image (by dragging from the white other frame) and dropping on the left image well labeled “Start Frame”, and dragging from mutation grid, up “through” the tab, and dropping it on the right “End Frame”.

Below that is a simple slider that is used to adjust how far “between” these two images we want, and below that is the larger “Current Tween” image showing the results. In this example, sliding from the left to right will slowly change the image from the blue background to the gold background one. This result can then be dragged and dropped as desired (in the current image, as a snapshot, added to the movie timeline, or any of the other drop targets).

Note that tweening doesn’t always create an image that seems like it is “between” the two images. There can be several reasons for this. One of the problems is that discrete style settings don’t have “between” values (and tweening only will generate “smooth” transitions). Another issue is that the final tweened value is based on the current image. Effectively, the parameters and colors that are calculated as being between the start and end frame (at a position based on the slider) are applied to the current image to create the current tween. If neither the start nor end frame have anything in common with each other or the current document, there is nothing that is actually “between” them (much like asking “what is halfway between a fire truck and a lamp post”).

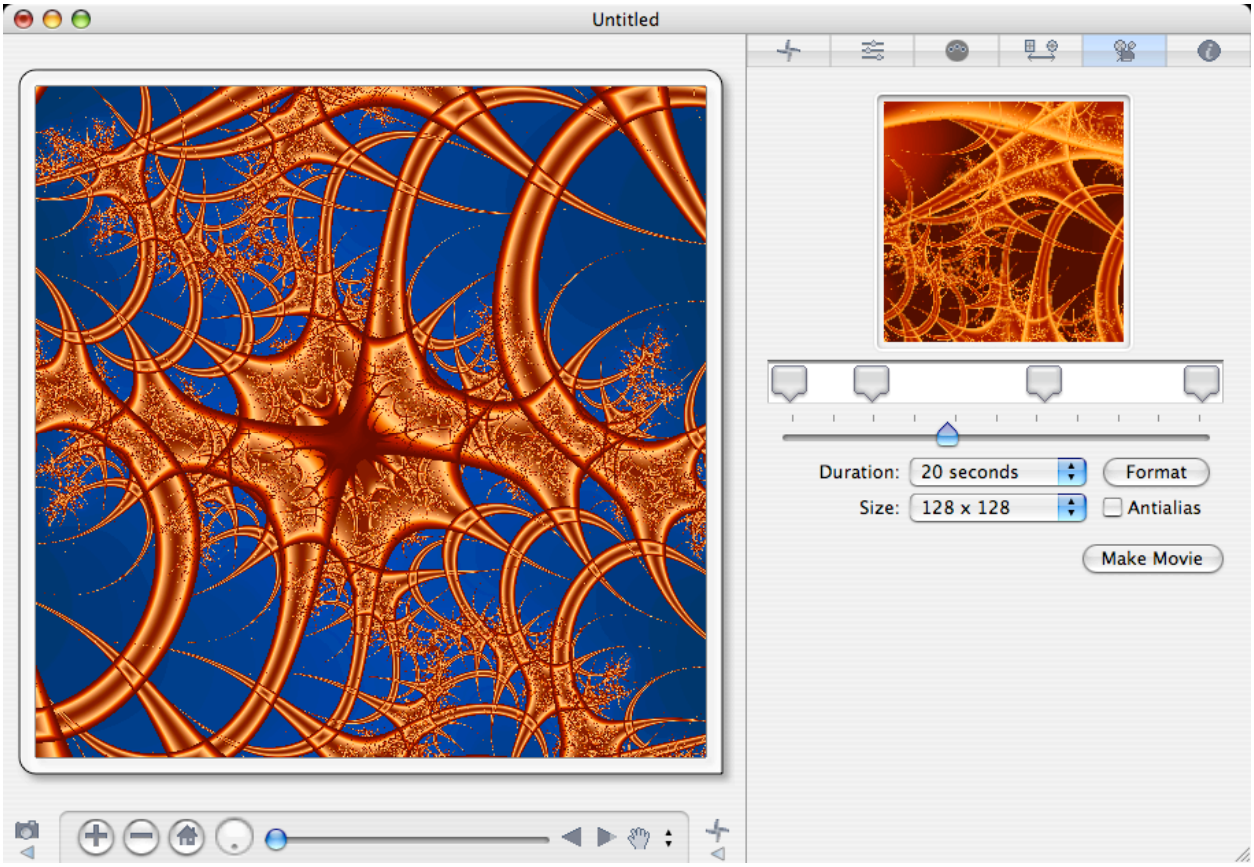
Tweening is also not restricted to just changing the parameters and colors - it will also move between different points of view. This includes panning, zooming, and rotation, allowing you to “fly through” a fractal, for example.

For best results, use the current image as the start image, and a mutation (or similar snapshot) as the end frame. However, interesting and unexpected results can be obtained when using other images, so don’t be afraid to experiment.

At the bottom of the pane is a button marked “Make Movie”. This will render a movie to disk that effectively shows what happens when you slide the slider from the left to right position. This creates a simple “morphing” movie with two “key frames” (the start and stop). It also works well for making “fractal fly through” movies (where you start in one location and then zoom in a great detail to show recursive nature of a fractal). For a more complex movie with multiple steps in it, use the movie tools found in the next tab.

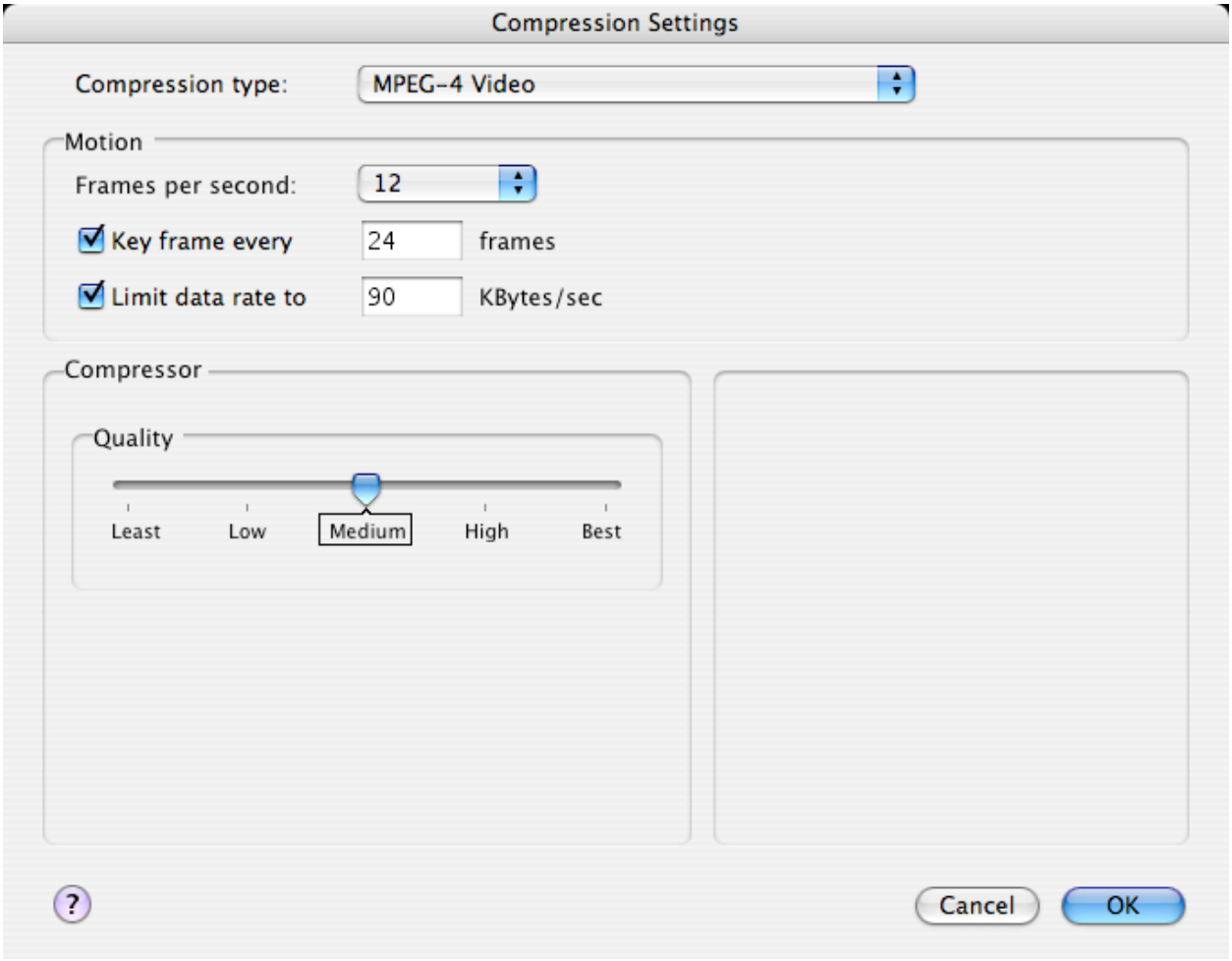
Creating Movies

The tweener can make a simple two frame movie, but you often want to have mulitple frames (”go from this image to this one, and then to a third, and back to the first”). You could create a series of movies using the tweener and then stitch them together, but that would be too much work. Instead, quadrium can generate multiple frame movies using the movie tools found in the fifth tab:



The movie panel is effectively a series of tweeners - below the image is a timeline showing the location of a series of key frames (the shiny rectangles). This works like a series of tweening (the two use identical techniques to generate their images). Unlike tweening, you can have multiple images at different “spacing”.

This pane also allows you to set the properties of the movie (these properties will also be used by movies made in the tweener pane). You can specify how long the total move will take, and how large it is. You can also enable anti-aliasing to create smoother looking images (though can take longer to render the movie). Most importantly, clicking on the “Format” button will bring the Quicktime video compression settings dialog:

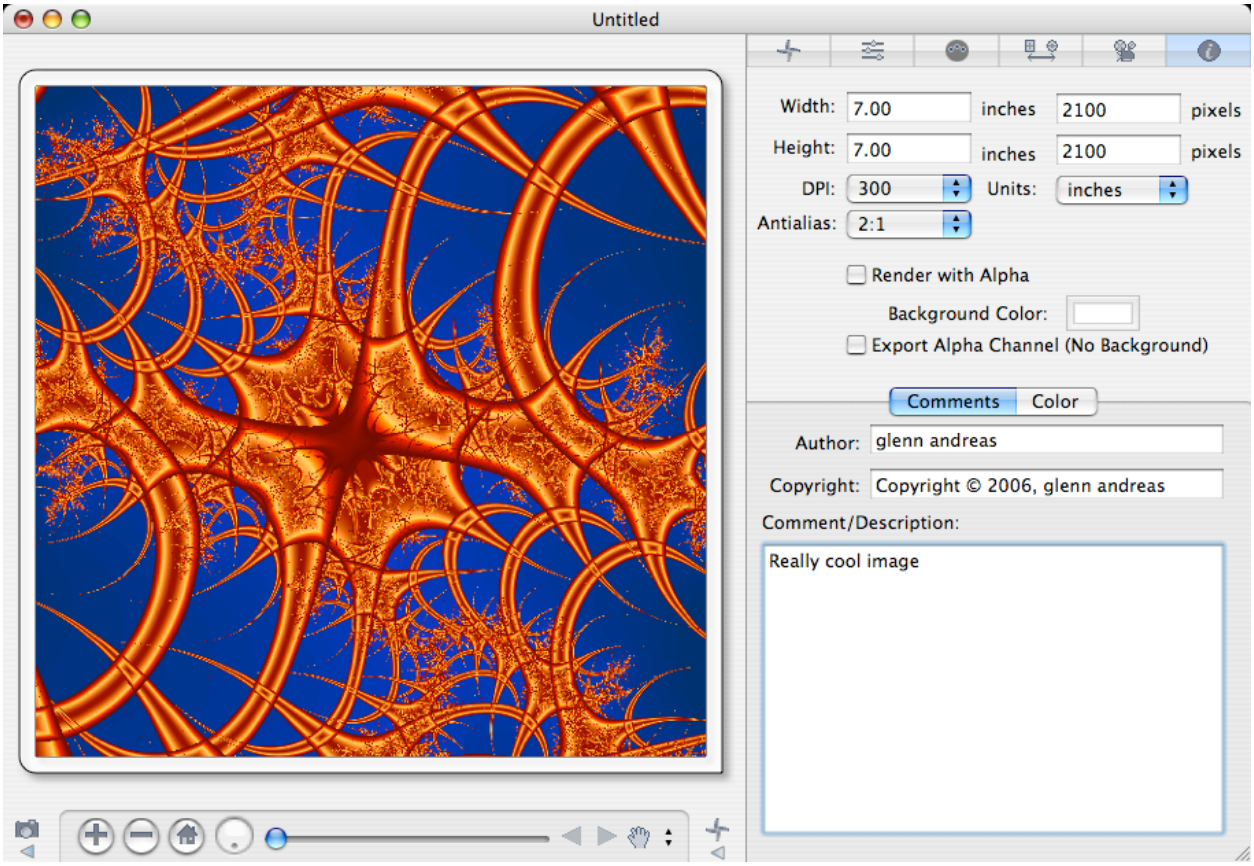


This is where you will specify what format you want the movie in (if you try to create a movie without specifying this, this dialog will be shown before you can save the movie). By default, it will be set to “video” compression - this provides very high quality movies, though the results will be fairly large. If you plan to share your movie over the net, you probably want to specify a format that compresses better (at the price of worse quality). MPEG-4 is a standard format, though many of these formats are designed for “real world” images and so will often produce poor results for computer generated images

(”Motion JPEG A” and “Motion JPEG B” produce good results, but can be ten times the size of MPEG-4 movies, but somewhat smaller than “Animation”).

Document Information

The final panel is used to set document specific information, “meta” information, and ColorSync adjustments:



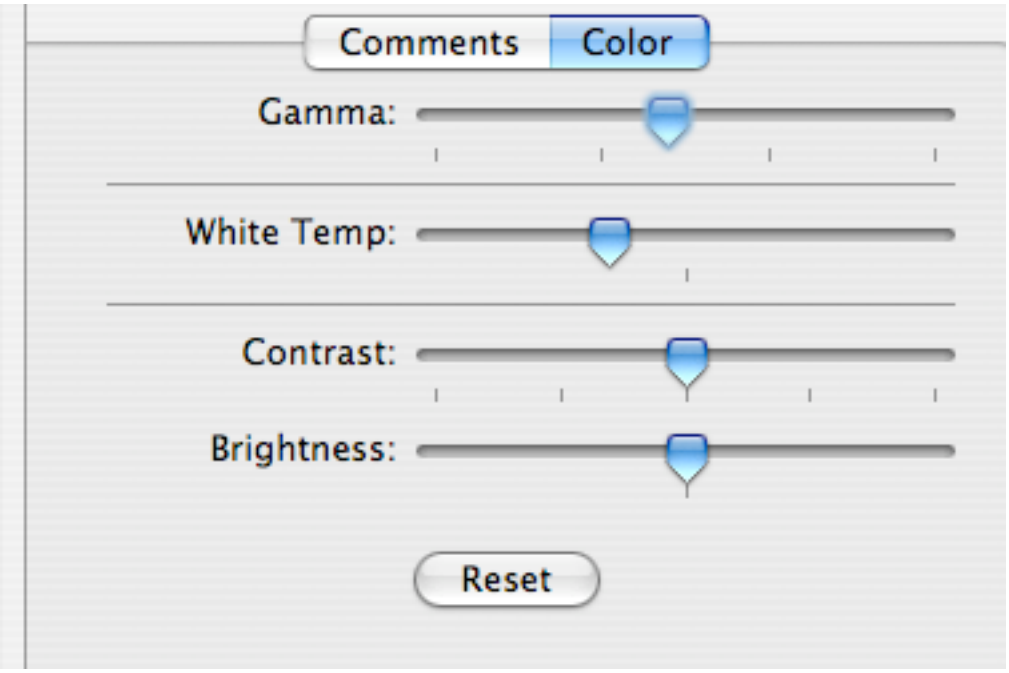
The top part of this has to do with the image itself. The primary part is how big the rendered/printed image is going to be - by default, it is 7” x 7” at 300 DPI (and 2:1 anti-aliasing), creating a very nice image that can be printed on a wide variety of printers or further manipulated in other programs. A variety of resolutions are supported - for web based work, you’ll probably want to pick 72 DPI.

The “Antialias” popup allows you to specify how much anti-aliasing to perform on the image. This will increase the amount of time it takes to create the image when rendering to file or printing (by a factor of 4 for 2:1, and 16 for 4:1) but can dramatically improve the resting image. This does not have any impact on the display of the image (2:1 anti-aliasing can be enabled for the display in the preferences).

Below that are settings that control how the image’s alpha channel is used (that fourth input nub in the “<Output Color>” node). By default, it is ignored and the image is solid. If you click the “Render with Alpha”, that channel is used to blend the image onto a

background, whose color you can specify here. This makes an easy way to fade an image in or out over a solid color (you can also feed in a solid color into a blender and set things up to use the blending alpha, but this is much easier). More importantly, if you then select “Export Alpha Channel” this means that when you render the image to file, it will not draw the background, but instead produce a file that has the alpha set accordingly. This allows you to create images that can be blended with other things in other applications.

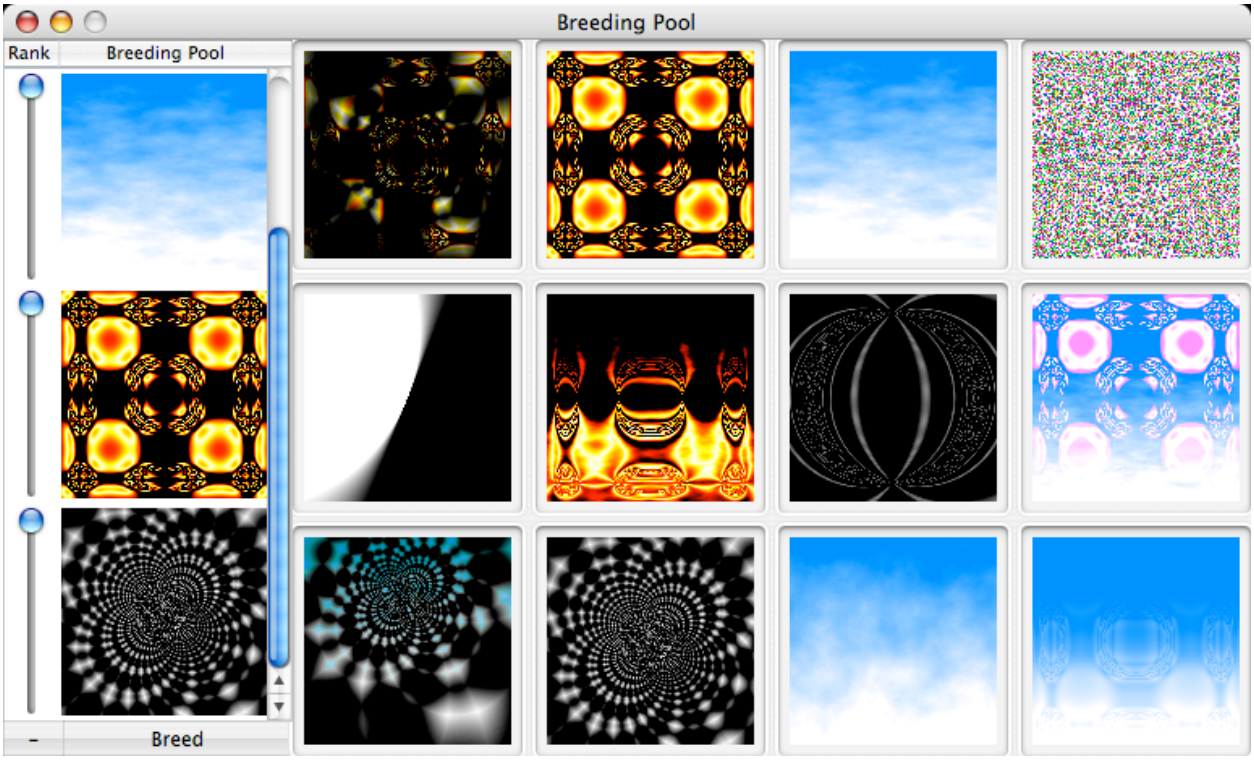
The bottom of this pane contains the “meta” information about the file, including the author (the current user), a default copyright, and a field where you can enter a comment or description (this shows up as a tool tip help tag in the starting points browser). All of this information is also available for search via Spotlight if you are running 10.4. You can also switch to the color matching information - images use ColorSync to help guarantee that the image you see matches the image that prints out on the printer.



This second sub-tab includes the ability to set the image’s gamma (defaulting to the standard Macintosh 2.2), as well as a white point temperature (6500°K) and adjust the contrast and brightness of the image. This information is all recorded into the ColorSync profile that gets embedded when you save the image.

Breeding

Breeding is a process of taking image and combining them to create new images. Conceptually, imaging taking the structure of an image as shown in the structure detail panel, and then slicing it across one of the nodes, cutting it in two. Take a second image and perform the same thing. Now take the top of one structure and connect it to the bottom of the other. The result is a new image - just like DNA genes can be combined like this, so too can quadrium images. It is controlled in a separate window from the document, shown by selecting **Window > Breeding Pool**:



There are two primary parts of the window - the pool of possible parents, and a collection of children (these will initially be blank). In the example above, three different images are in the breeding pool. Two images are picked at random from this breeding pool, and their structures are sliced and combined to produce a new image. These “children” are then shown on the right. You can see a variety of images, some of which are nearly identical to some of the parents, others are clear blend of two of them.

To create a new collection of children, clicking the “Breed” button will generate a new series of images. To help to select which images are used to breed a new generation of images, each potential parent has a “rank” associated with them. The higher the rank, the more likely it will be chosen to parent a new image. This works well when you’ve got some images that you like better than others.

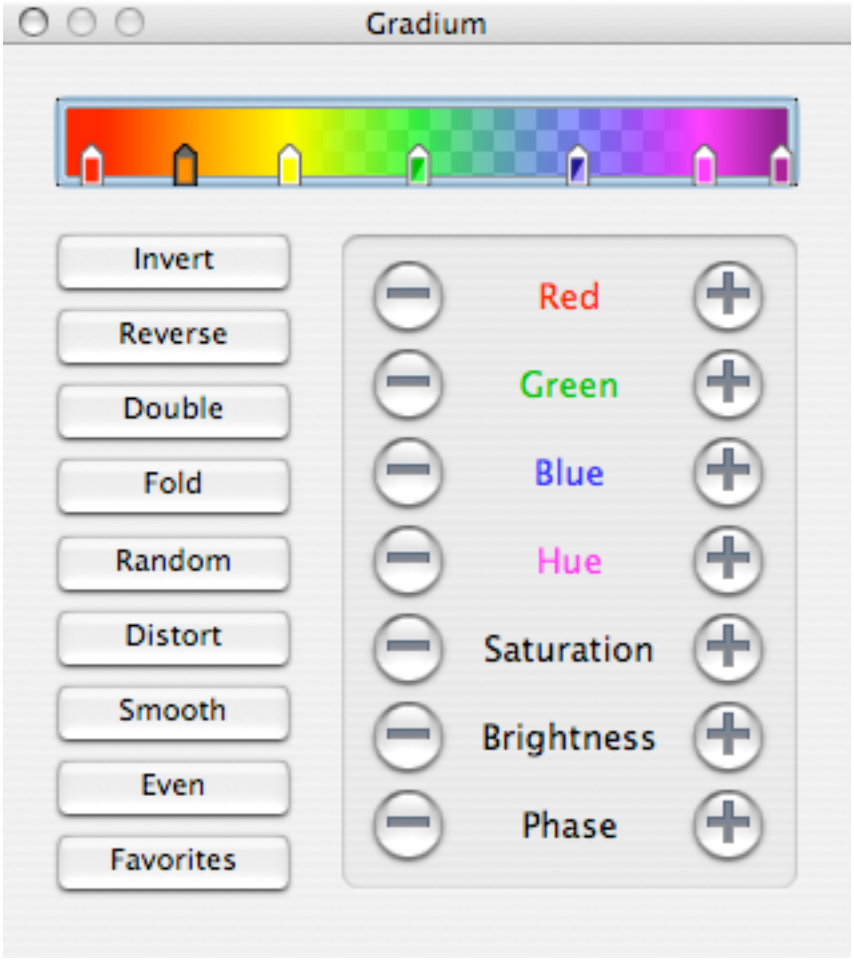
To add images to the breeding pool, drag them to the “Breed” drop target below the menu bar. To remove images, select them and, click the “-” button. You can drag im-

ages from the children and drop them in any drop target (such as “New” to create a new document using that image, or even “Breed” to add the child to the breeding pool). Like other images thumbnails, you can click on them with the control and option keys held down to popup a larger view of the image.

Besides the “slice and combine” technique described above, there are a number of other possible ways that the images can be combined, such as putting them in parallel and combining the two images with a “Blender” node. Like DNA reproduction, there is also a chance of mutation occurring, creating a new image completely unlike any of the possible parents. Note that repeated use of breeding can produce very complex images (such as some of the images found in the “Monster” starting point category), and as a result, can produce images that may tax some of the most powerful machines, so some restraint should be used to avoid this possibility.

Working with Gradients

The use of a gradient is common in quadrium - it is used to convert a single number into a range of colors. For example, the numbers 0 through 1 could be different points on a rainbow, so 0 would correspond to red, 0.5 would correspond to green, and 1.0 would correspond violet, with a smooth continuum of colors between them (these colors may also have different alpha values). To make it easy to work with these gradients, when you select one, the gradient editor floating palette is shown:



This panel functions much like the color picker panel works - when a color is selected in a color well (i.e., the border is highlighted), changes to the color in the color picker will result in changes to the color in the document. Similarly, when a gradient is clicked on, it becomes highlighted and changes to the gradient in the gradient editor changes the gradient in the document. If you have more than one document open, only the active document's gradient should be changed.

As you can see from the image here, the gradient editor shows the current gradient, which is annotated by markers that show where specific colors are. These markers can be clicked on (allowing the color to be edited via the standard color picker panel, including changing the alpha channel, as demonstrated by both the green and blue markers in the illustration), moved around, reordered, or deleted by dragging them off the gradient entirely. Below the gradient are two groups of controls - one a set of actions (the buttons on the left), the other a set of adjustments. Looking a the actions on the left:

- **Invert** - inverts all the colors in the gradient - black becomes white, red becomes cyan, etc...

- **Reverse** - swap the order of the colors in the gradients, so this example would start with purple.
- **Double** - repeats the gradient a second time, appending them together and compressing them to the original gradient, so we would have two sets of rainbows in this example.
- **Fold** - Fold combines “Double” and “Reverse”, repeating the gradient after reversing it. This would result in a gradient that went from red to purple and then back from purple to red.
- **Random** - Randomizes the colors in the gradient, but does not change the location of them.
- **Distort** - Reorders the locations of the colors in the gradient, mixing it up (but maintaining the same colors)
- **Smooth** - Smoothes out the gradient so that changes are more gradual
- **Even** - Redistributes the locations of the color markers to be evenly distributed.
- **Favorites** - Toggles a drawer allowing you to pick from a large set of preset gradients (among a number of categories), or save your current as a favorite for later use.

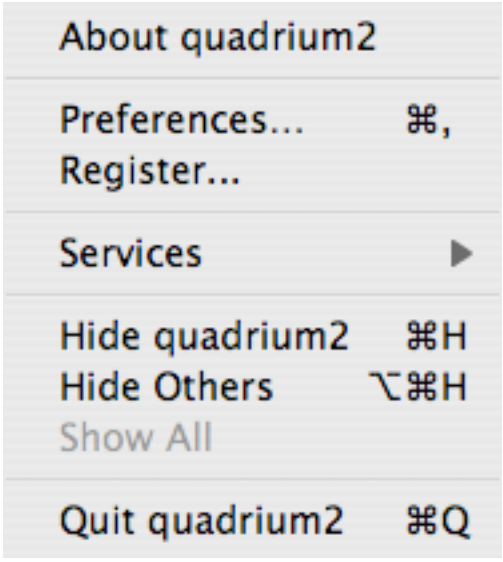
The controls on the right will adjust all the colors of the gradient. They are as follows:

- **Red** - Increases/decreases the amount of red in all the components
- **Blue** - Increases/decreases the amount of blue in all the components
- **Green** - Increases/decrease the amount of green in all the components
- **Hue** - Adjusts the hue of all the colors, shifting through reddish/greenish/blueish or back.
- **Saturation** - Adjusts the saturation of all the colors, making them more vibrant or washed out.
- **Brightness** - Adjusts the brightness of all the colors, making them lighter or darker.
- **Phase** - Will shift the gradient left/right, wrapping around to the other side. Unlike the other adjustments, this will actually move the markers.

Menus

The menus found in quadrium are as follows:

The quadrium2 menu, besides items common in most application, contains:

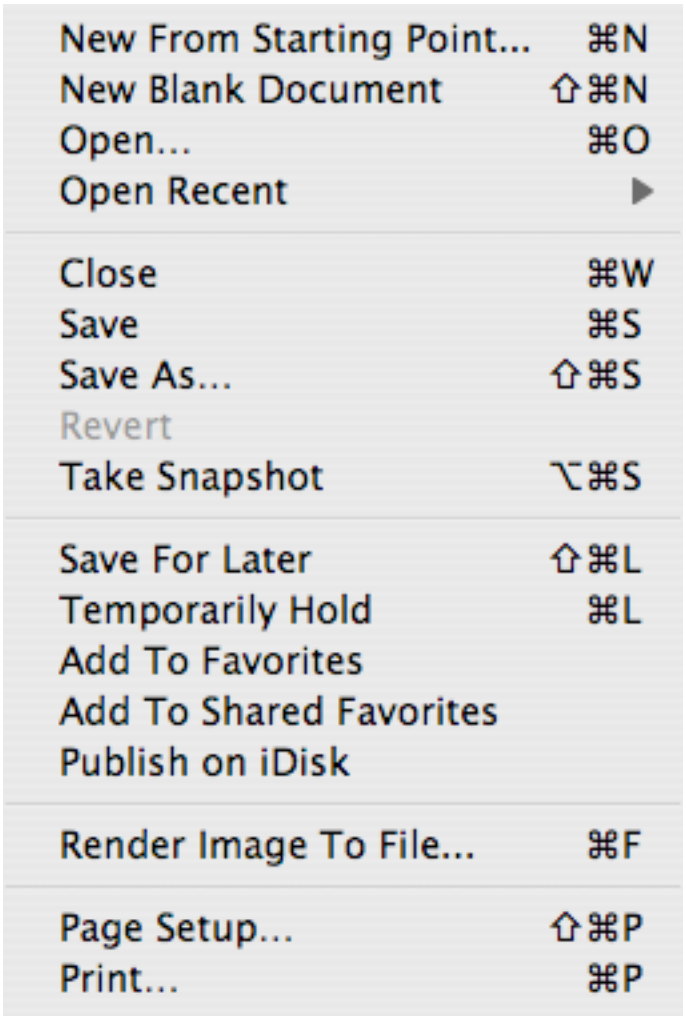


• **About quadrium2** shows the version, copyright and credits.

•**Preferences** allows you to set the application preferences as detailed in the “Preferences” section below.

•**Register** allows you to register your copy of quadrium. Until you register, rendered and printed images and movies will all have a the word “demo” watermarked on them, and you will be unable to save the file. If you hold down the option key, this will change to “reinstall”, allowing you to reinstall your copy if something happens to your machine. Note that you will need an active network connection to be able to complete this process.

The **File**, besides items common in most applications, menu contains:



• **New From Starting Point** will display the starting points browser

•**New Blank Document** creates a new image with the input directly hooked up to the output and no additional nodes in the document. Useful if you want to construct an image entirely by building it up node by node.

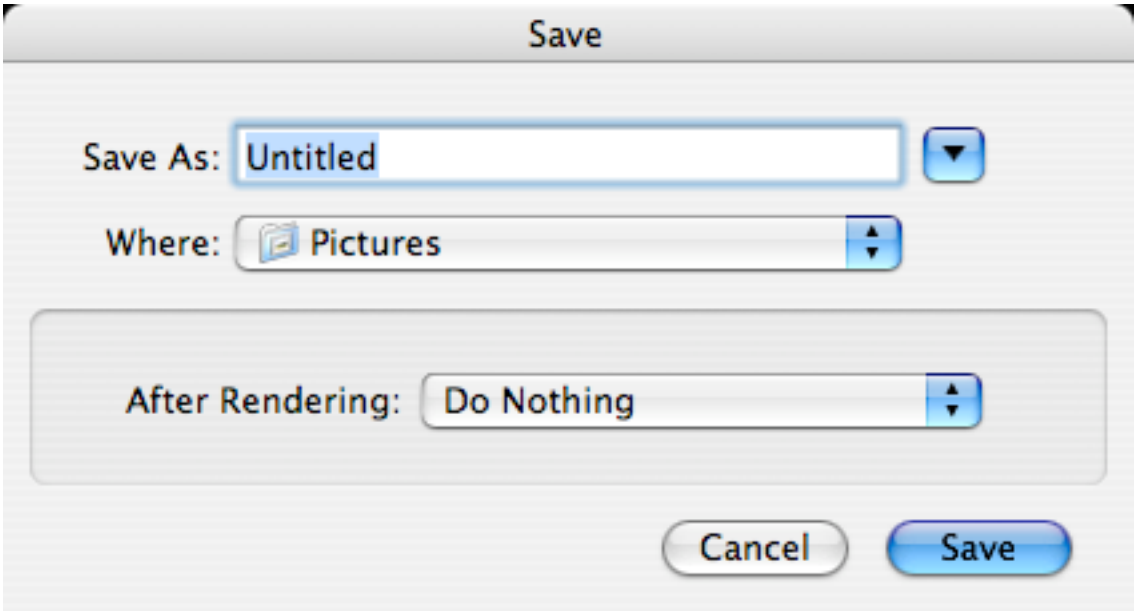
•**Take Snapshot** is a keyboard/menu shortcut to add the current image to the list of snapshots in the document.

•**Save For Later** is the same as dragging the image to the “Later” drop target, adding it to the (persistent) category “Saved for Later”.

•**Temporarily Hold** is the same as dragging the image to the “Temp” drop target, adding it to the temporary category.

•**Add to Favorites** adds the document image to the favorites starting point category (same as dragging it to the “Fav” drop target).

- **Add to Shared Favorites** adds the document image to the locally shared favorites (same as dragging it to the “Share” drop target)
- **Publish on iDisk** adds the document to your iDisk for others to see (same as dragging it to the “iDisk” drop target).
- **Render Image To File...** will prompt you for a file to save to and then begin rendering the image at the resolution and size specified in the document information tab, as described above. When rendering to a file, you also have an option to specify what action to perform when the image has been rendered:

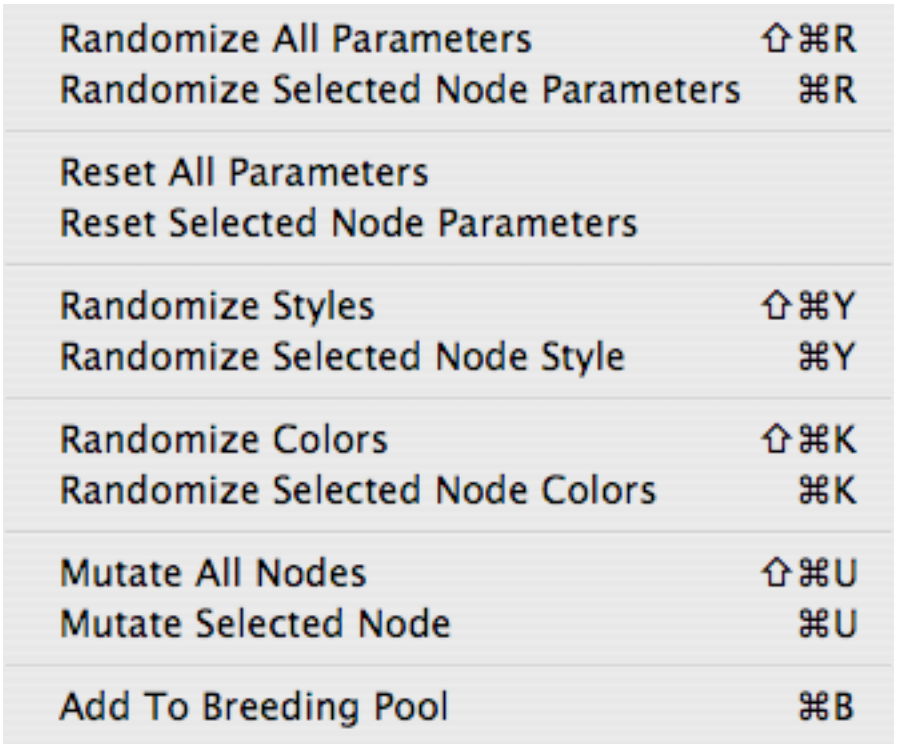


This menu includes (besides “Do Nothing”) the ability to open the image file in various applications (quadrium will ask the system to give it a list of files that will open the image file), as well as a possible list of “file actions” (AppleScript scripts or Automator workflows as described in the “Rendering Actions” section below).

- **Print...** will print the image, again at the size and resolution specified. Because both printing and rendering to file are time consuming processes (especially with high anti-alias factors at high resolutions), this is all done in the background (as is rendering movies). A floating job window will appear showing you the progress of these tasks.

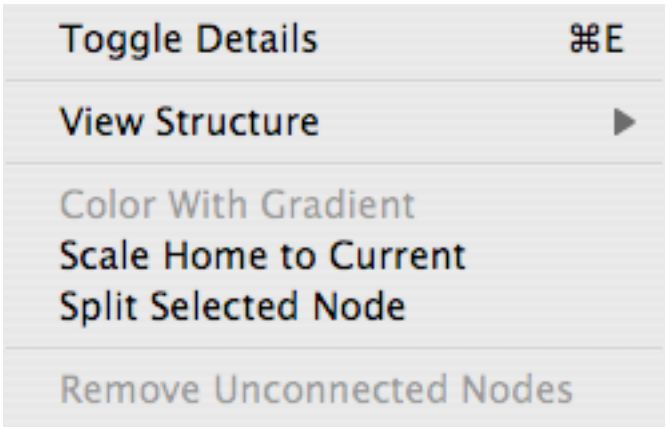
The **Edit** menu is a standard edit menu. quadrium, however, allows you to hold down the shift key to modify the behavior of “copy”, depending on the currently active UI element (for example, “copy” can become “copy image with alpha” to copy the current view image with alpha channel).

The **Random** menu contains the following commands, which act as a small subset of the “mutation” tab:



- **Randomize All Parameters** cause all (unlocked) parameters of all nodes to be set to random values
- **Randomize Selected Node Parameters** only randomizes the (unlocked) parameters of the currently selected node.
- **Reset All Parameters, Reset Selected Node Parameters** resets the parameters of all or the selected node respectively.
- **Randomize Styles, Randomize Selected Node Style** randomizes the style settings of all or the selected node respectively.
- **Randomize Colors, Randomize Selected Node Colors** randomizes the colors of all or the selected node respectively.
- **Mutate All Nodes, Mutate Selected Node** will change all the nodes or only the selected node. These nodes will be replaced with “similar” nodes.
- **Add to Breeding Pool** provides a menu/keyboard shortcut to add the current image to the breeding pool, the same as dragging the image to the “Breed” drop target.

The **Structure Menu** contains the following commands:



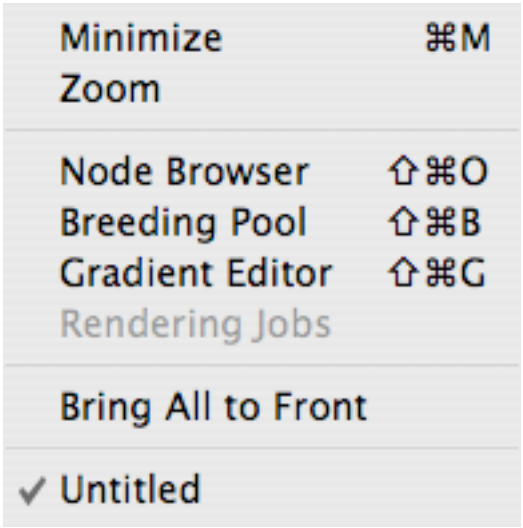
• **Toggle Details** will show/hide the details pane, automatically selecting the first (structure) tab

• **View Structure** submenu allows you to change how to display the structure (name, small icon, large icon)

• **Color with Gradient** - if the current image is grayscale (all channels of the output color come from the same node), this command will insert a gradient node between that node and the color output node. This makes it easy to use a gradient to provide color for the image.

- **Scale Home To Current** - This will add a node that cause "Home" location to be where the current view point is (by having that node perform the scaling, translating, and rotation to match what you did).
- **Split Selected Node** - Will take the currently connected node and split it in half, making an exact copy with the same inputs (but no connected outputs).
- **Remove Unconnected Nodes** - If there are nodes whose output is not connected to the final output node, it is removed (this will not change the appearance in anyway, but is useful to help clean up the structure).

The **Window** menu, besides items common in most applications, contains the following:



• **Node Browser** will show the floating node browser window

• **Breeding Pool** will show and select the breeding pool window.

• **Gradient Editor** will show the floating gradient editor window.

• **Rendering Jobs** will show the floating window listing the current background rendering jobs (rendering to file, printing, or generating movie), allowing you to check their progress or cancel a specific job.

File Actions

One of the powerful features of quadrium is that it allows you to work image generation into your existing tools workflow. To do this, it provides the ability to add various actions to perform when a file is rendered. quadrium automatically provides a list of applications that the system is aware can open image files. You can specify additional applications in the preferences panel in the “File Actions” pane by clicking “Add” and selecting the application.

However, quadrium allows you to create AppleScript files or Automator workflows (which require 10.4). To use AppleScript, you need to write a script that contains a routine named “processImage” that takes a single parameter, which is the path to the file. In that routine you can write whatever actions you wish to have happen. For example, to set your desktop picture to newly rendered file, you could have the following script:

```
on processImage(filepath)
	tell application "Finder"
		set desktop picture to filepath
	end tell
end processImage
```

If you save this script to a file (such as “Set Desktop Picture”) and then select that file after clicking the “Add” button, it will then appear in popup menu the next time you render a file. If you select “Set Desktop Picture”, after the image is rendered, the script will be executed which causes the desktop picture to be set.

To add it to iPhoto in an album named “quadrium” you could use this script:

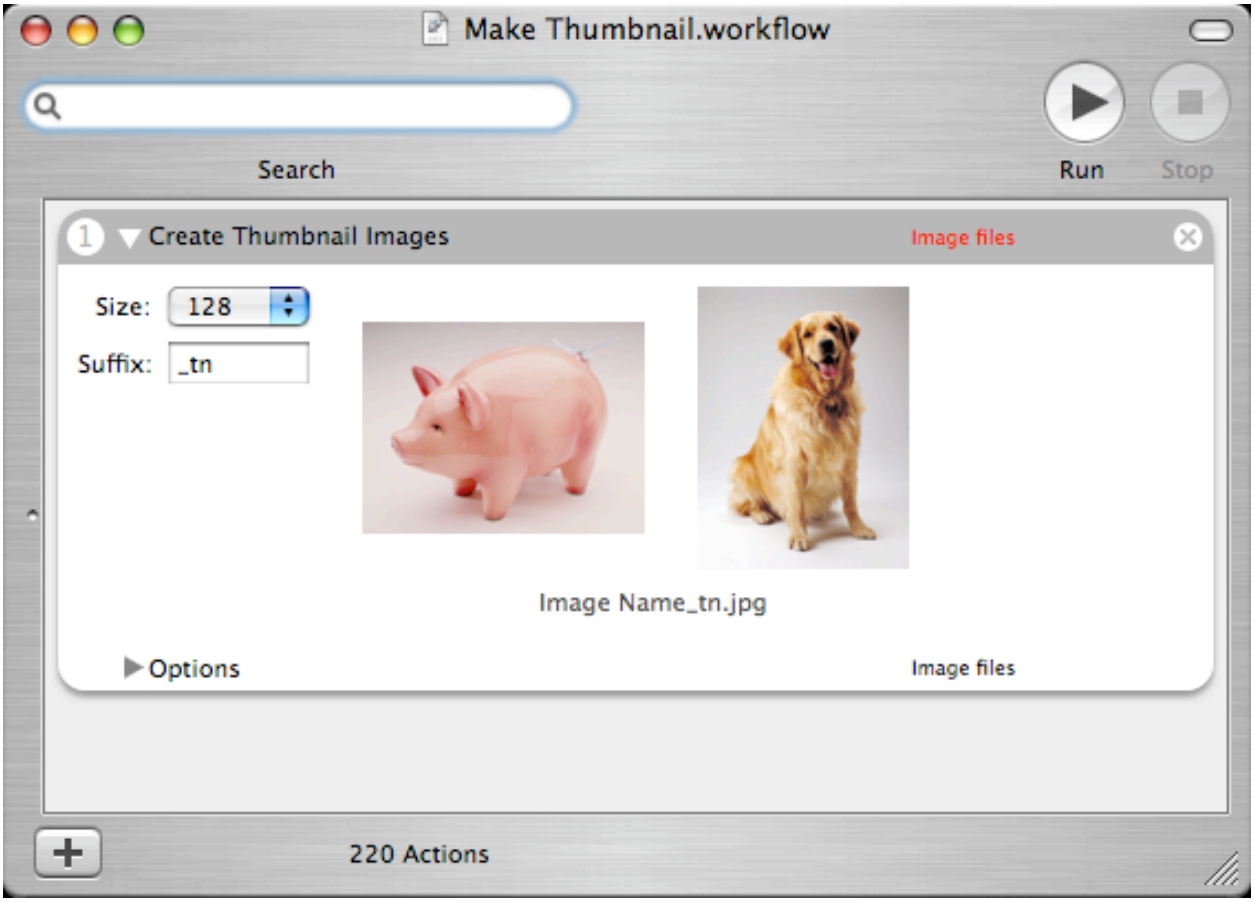
```
on processImage(filepath)
	tell application "iPhoto"
		-- create the album if needed
		set qalbum to false
		try
			set qalbum to album "quadrium"
		end try
		if qalbum is false then
			new album name "quadrium"
			set qalbum to album "quadrium"
		end if
		-- now add our image
		set current album to qalbum
		import from filepath to qalbum
		activate
	end tell
end processImage
```

To send that image as a mail message, the following script would work:

```
on processImage(filepath)
	tell application "Mail"
		set newmessage to make new outgoing message with properties
				{subject:"Cool quadrium Image", content:"Isn't this im-
				pressive?" & return & return}
		tell newmessage
			set visible to true
			tell content
				make new attachment with properties {file name:filepath}
					at after the last paragraph
			end tell
		end tell
		activate
	end tell
end processImage
```

Note that these scripts are available in the “Image Scripts” folder included with quadrium.

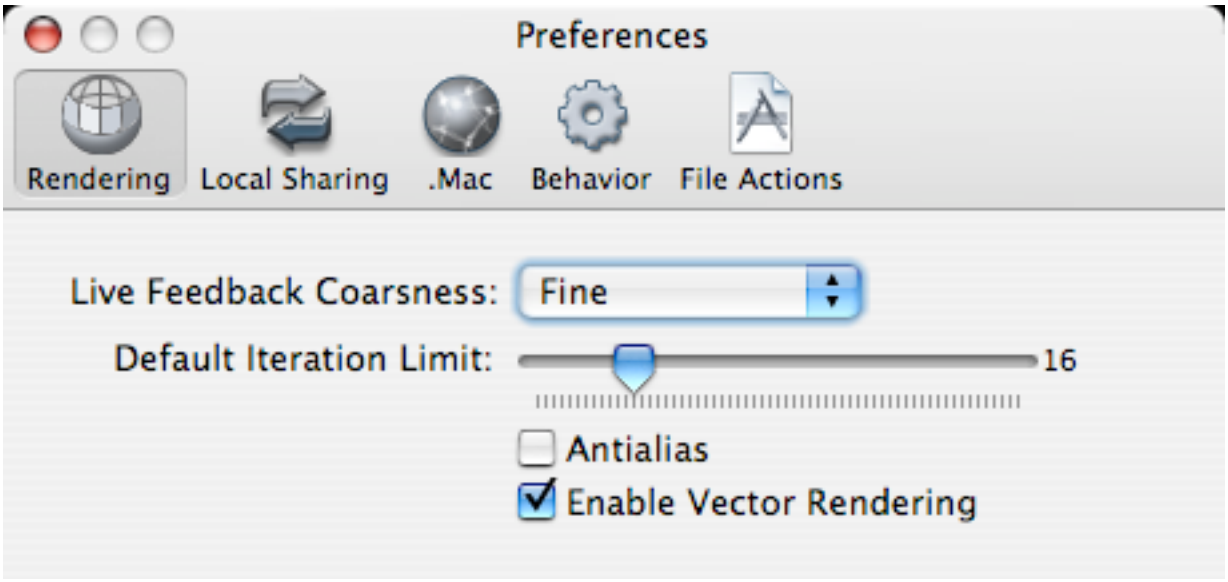
Using an Automator workflow is even easier. Just make a workflow that has “Image File” as input. For example, to generate a thumbnail, a workflow that looks like this would do the job:



(This specific action comes from the actions found in “Preview”) You would select the “Make Thumbnail.workflow” file just like an AppleScript script file (if the workflow isn’t saved as an application, quadrium will prompt you to automatically do this for you), and it will be added to the list of available actions.

Preferences

quadrium contains a variety of user preferences, selected from the **quadrium2 > Preferences** menu item. This preference window is divided into the following panes:



Rendering

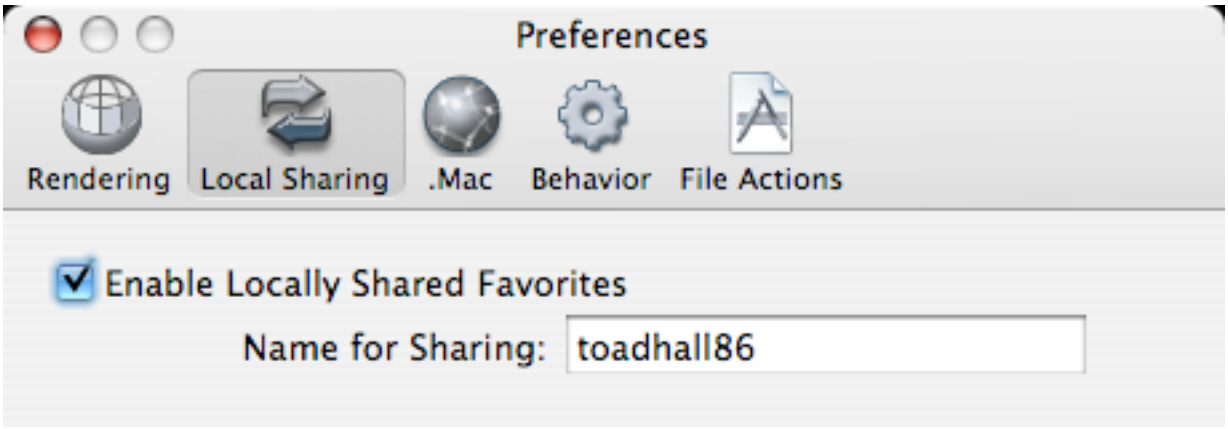
Controls the various options for how the image is displayed on screen (rendering to file is handled in the document info tab in the details pane). The most important is the menu that allows you to change how coarse the live feedback is drawn at. Because live feedback is an important part of the quadrium user experience, you can adjust how detailed the image is while you are making changes to things such as slider - the coarser it is set, the faster it will respond (but the “chunkier” the image will look).

Default iteration limit is used to determine how many times to iterate things such as escape fractals (nodes that use this also provide a parameter that lets you change it to a different value - this is used to provide an initial default value). Setting the value high will cause adding certain nodes to potentially slow down the responsiveness of quadrium - setting the value too low will cause these images to look very simplistic and boring (and the point of fractals is often in the detail). Note that changing this value does not change existing nodes, only new ones.

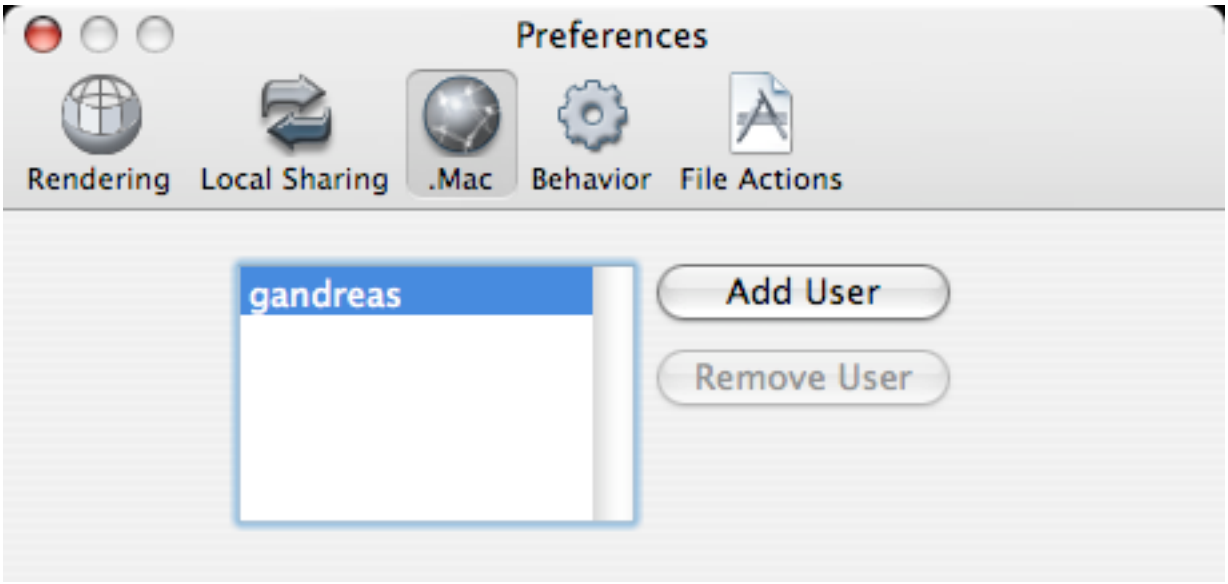
Antialias checkbox allows to perform 2:1 antialiasing in the main view, which greatly improves the appearance of images (especially with fine detail) but slows it down by a factor of up to 4. Again note that this is only for the on-screen window appearance - actually rendering is controlled by the document settings.

Enable Vector Rendering is a checkbox that should normally be set (if possible). On G3 based PPC machines, this is disabled (because there is no vector unit available). Turning this off will slow down the responsiveness of the display, and should only be done if there are strange drawing anomalies (or if you really want to see the difference between

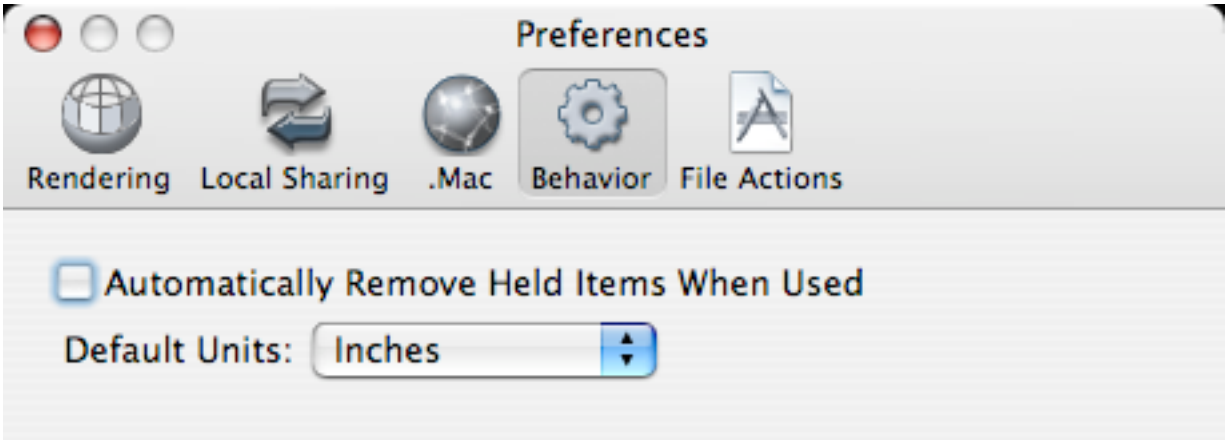
Altivec and non-Altivec performance). Turning off vector rendering can also help reduce the amount of power consumed on laptops to extend battery life.



Local Sharing
This pane allows the user to enabled or disable sharing images locally with other users on their (local) network. You can also alter the name that appears in the starting points browser under the “locally shared” category. Note that disabling this will not effect being able add a file to that category, just if anybody else can see it.

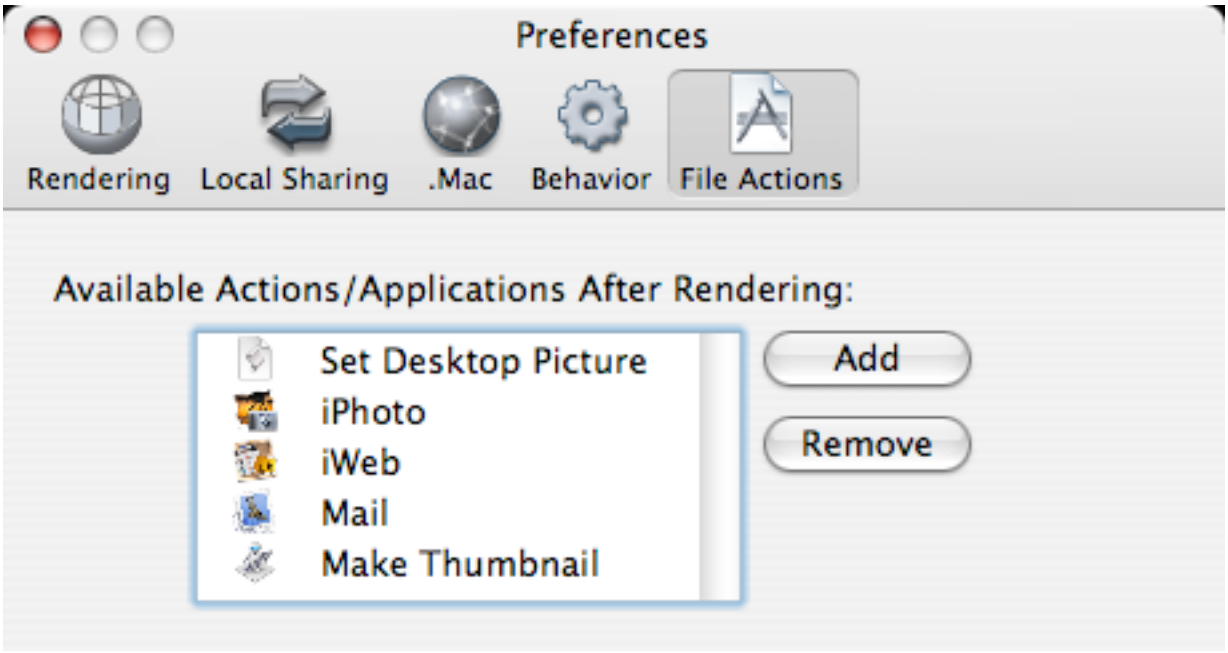


.Mac preferences
This pane is used to configure sharing images globally via the users iDisk, and whose iDisk is visible in the starting points browser. By default the iDisk for gandreas software is always on this list. You can add another user by clicking the “Add User” button and entering their name in the dialog.



Behavior
The behavior pane is used to control some basic program behavior. There are two options here - the first is a checkbox that allows you to automatically remove “held” items. When you either “hold for later” or “temporarily hold” an image, it is added to the appropriate category. When you select it from the starting points browser, quadrium will ask if you want to remove it from that category (since you wanted to put the image “on hold” for a while, if you are now getting back to it, you often don’t want to keep it in that list). With this check box checked it will automatically remove it without prompting (how you use this feature depends on your personal workflow).

This pane also allows you to specify if you’d prefer to work in “inches” or “centimeter” for measurement (both in terms of the document size, as well as the ruler)



File Actions

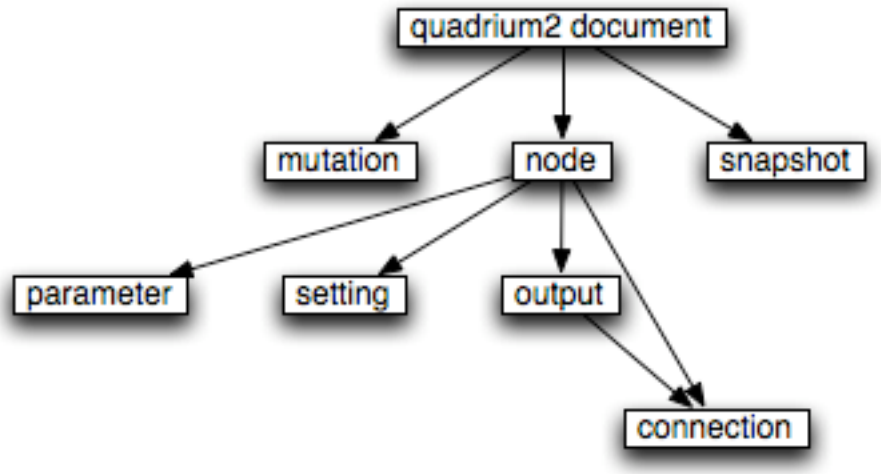
As discussed above in the “File Actions” section, you can provide a list of applications, workflows, or AppleScripts to be performed on a rendered file. This pane allows you to edit this list. Clicking the “Add” button will let you specify an application, AppleScript or Automator workflow (Automator requires 10.4). To remove a selected item, click the “Remove” button. These items will then appear in the file saving dialog shown when **File > Render to File** is selected, under the “After Rendering:” popup menu.

AppleScripting quadrium

Starting with quadrium2.1, you now have the ability to write AppleScript scripts to automate various actions. These can range from manipulating the nodes, parameters and settings of the current image, to working with snapshots or rendering new images. The whole AppleScript dictionary can be seen using Script Editor's "Open Dictionary" command (quadrium provides a menu item to launch Script Editor, especially handy for those of you who don't always remember where it is located).

When quadrium2 is launched, it automatically looks through the folder called "Scripts" in your ~/Library/Application Support/quadrium2 folder (which can be easily shown by selecting the "Show Scripting Folder" menu item). All scripts that it finds there will automatically be added to the scripting menu (located as a submenu in the "Structure" menu). You can organize your scripts with additional folders - these will get turned into submenus accordingly.

The first thing to know about writing scripts is to understand the correspondence between the various items in quadrium2 and objects in AppleScript. The graph below shows the relationship between the various objects:



So the quadrium2 document contains mutations, snapshots, and nodes. Nodes have parameters (which correspond to those values set with sliders), settings (which are set by various assorted user interface elements such as checkboxes, radio buttons, and popup menus). The nodes then represent the structure of the document using outputs and connections - the outputs are on the bottom of the node, and the connections represent the top of the node. Notices that the output also contains a list of connections - this is because a given output can connect to multiple inputs.

To create a new node in the document:

```
set adjNode to add node named "Value Adjuster" after
    gradientNode
```

(where ‘ gradientNode’ is a reference to an existing node in the document - this is also assumed to be inside a tell block such as “tell first document”). There are several variations - the example above adds the new node after an existing node - you can also use ‘before’, ‘split’, ‘replace’, or nothing at all. The name of the node is the name as shown in the node browser (special characters such as accents or superscripts are treated as their plain equivalents)

To “wire” something up:

```
connect fourth output of fractalNode to fourth connection of
        adjNode
```

(where ‘ fractalNode’ and ‘ adjNode’ are both nodes in the document). Note that you always specify the output and connect it to a connection of another node, numbered from left to right.

To set a parameter:

```
tell adjNode to set value of (parameter named "midpoint") to 0.0
```

Setting a parameter is basically a two step process - you need to tell the node to set a specific parameter’s value, so you need to access the correct parameter using the ‘named’ property (all parameters have names). These names are based on the text displayed next to the slider, with spaces removed and converted to lowercase characters. Besides setting the name of a parameter, you can also set the “locked” property (allowing to you script locking or unlocking multiple parameters).

To change a setting:

```
tell adjNode to set value of (setting named "operation") to 2 --
        "scale"
```

This is similar to setting a parameter - again, you need need to access the setting by name (the exact name varies between nodes for historical reasons). To find out all the settings of a node, you can do something like this:

```
name of settings of adjNode
```

(You want “name of settings” not “names of settings”). This will return a list of names.

The value you set it to will also vary - often it is an integer, and the correspondence between values and the user interface varies from node to node. The safest way to set these values is to first set them in the user interface, and then use something like:

```
tell adjNode to get value of setting named "operation"
```

Again, a two step process.

Finding nodes in documents:

```
set fractalNode to first node whose kind contains "fractal"
set gradientNode to last node whose name contains "gradient"
```

There are two strategies for find specific nodes in an existing document. You can search through to find a node by name (using the "whose name contains" clause) or by kind (using the "whose kind contains"). Kind corresponds to which category the node is found in node browser. You could also walk through the node structure explicitly, either from the top (starting at the document's "input node" and walking through the node's "outputs") or at the bottom (starting at the document's "output node" and walking through the node's "connections"). For example:

```
set downstream to {}
repeat with o in outputs of n
    repeat with c in connections of o
        if downstream does not contain {destination node of c}
                then
            set downstream to downstream & {destination node
                    of c}
        end if
    end repeat
end repeat
```

Obviously, walking through an image's structure like this is rather convoluted.

Manipulating the current point of view:

```
set rotation of its view space to 0.5
set center x of its view space to 2.3
set center y of its view space to 1.5
set scale of its view space to 10
```

All of these actions will change the view of the appropriate document. Rotation is given in "half revolutions", so 0.5 corresponds to 90°.

Common Document Operations:

```
tell first document to scale to home
tell first document to remove unconnected
```

These actions correspond to the similarly named menu options.

Additional Actions
There are a wide variety of additional actions that are supported, such as working with snapshots, being able to dynamically run the tweener (to allow you to script tweening snapshots), accessing starting points, rendering images to a file, etc... For a full list, please consult the scripting dictionary.

You might notice that besides the quadrium2 suite, the scripting dictionary includes an additional "quadrium math" suite - this provides a number of more advanced mathematic operations that aren't supported by AppleScript, such as trigonometric functions.