

# CFG

## Color Fractal Generator

# User's Manual

version 2.3

for Color Fractal Generator 2.3

by John A. Schlack

August 1995

## **Table of Contents**

List of Illustrations .....	v
Requirements .....	1
New in Version 2.3 .....	2
CFG Contents .....	2
Unregistered Version .....	2
Standard Version .....	3
Master Version .....	4
Very Brief Background .....	4
Detailed Fractal Description .....	5
Mandelbrot Set .....	6
Julia Set .....	7
Dragon Curves .....	7
Polynomial Class .....	8
Transcendental Class .....	

8	
Random Walk Fractals .....	
9	
Generating Fractals .....	
10	
Basic Concepts .....	
10	
Fractal Parameters .....	
12	
Selection .....	
18	
Zoom .....	
19	
Applying Color .....	
21	
Saving Data .....	
28	
Opening Fractal Files .....	
29	
Preferences .....	
29	
Program Preferences Dialog .....	
30	
Generate Immediately .....	
31	
Draw Individual Lines .....	
31	
Display Progress Scale .....	
32	

	Display Random Walk .....
33	Limit Window Size .....
33	Precision .....
34	Window Preferences Dialog .....
34	Destination .....
35	Save Fractal Image .....
35	Save Palette .....
35	Save Selection Only .....
36	Use Faster Direct Color .....
36	Depth .....
36	Apply All .....
36	Creator Preferences Dialog .....
37	Image Creator .....
37	Text File Creator .....
38	Fractal Size and Memory .....
38	Help System .....
39	Main Menus .....

42	
Apple	.....
42	
File	.....
43	
Edit	.....
45	
Fractal	.....
47	
Size	.....
48	
Special	.....
49	
Standard Palettes	.....
51	
Function	.....
52	
Parameters	.....
53	
Background Speed	.....
54	
Windows	.....
55	
Scripts	.....
55	
Pop-Up Menus	.....
57	
Open Palette	.....
57	
Depth	.....
58	
Scripting	.....
59	

Shared Menus .....	60
Speed Comparisons .....	61
Machines Tested and Bugs .....	62
PowerMacs .....	63
Registration .....	64
Feedback .....	66

## **List of Illustrations**

Figure 1: New Parameters Window .....	10
Figure 2: Random Walk Parameters Dialog Box .....	13
Figure 3: General Parameters Dialog Box .....	15
Figure 4: Transcendental Parameters Dialog Box .....	17
Figure 5: Polynomial Parameters Dialog Box .....	17
Figure 6: Fractal Scaling Dialog .....	20
Figure 7: Maximum Colors Dialog .....	22
Figure 8: Palette Editor .....	24
Figure 9: Load Palette Dialog .....	26
Figure 10: Resource Palette List Dialog .....	27
Figure 11: Preferences Dialog .....	30
Figure 12: Program Preferences Dialog .....	31
Figure 13: Progress Scale .....	33

Figure 14: Window Preferences Dialog .....	34
Figure 15: Creator Preferences Dialog .....	37
Figure 16: Custom Fractal Size Dialog Box .....	39
Figure 17: CFG Help Manager Window .....	40
Figure 18: Quick Script Dialog .....	60
Table 1: FPU/Color Depth Speed Comparison .....	61
Table 2: Generation Speed Comparison .....	62



## **Requirements**

Color Fractal Generator (CFG) is only intended for use on a Macintosh color computer with at least a 4 bit (16 color) display and requires System 7.0 or later. This program supports 4, 8, 16, and 32 bit color systems, but the monitor must be at least 12" in size (minimum of 512 pixels by 384 pixels). CFG does not require a math coprocessor (even though it is doing some heavy number crunching), but it will take advantage of one, if present. The program is 32-bit clean and can generate fractals in the background, permitting the user to engage in other frivolous (or important) activities.

This program has been designed to provide extensive scripting support. CFG has the appropriate code to handle both AppleScript and UserLand Frontier. Using the scripting capabilities is definitely not required to generate great fractals using CFG, but it does provide the user with the ability to extend the power of this program. One can write scripts to provide simple utilities that make CFG easier to use (such as the Stack Windows sample script), automate the exploration of the fractal universe, or even create fractal movies. If one does not possess an AppleScript extensions or the commercial version of Frontier, Frontier Runtime may be used. This is the Shareware version of Frontier. It has some limitations, but it does provide a lot of the capabilities of the full version, but at a greatly reduced cost (since it's Shareware, you can also try before you buy). The runtime version is available on bulletin boards or Shareware CD-ROMs. If you have access to the full version, even better.

Fractals added in version 2.3 are very computationally intensive. It is suggested that the user have a high end 68040 or a PowerMac for generating these fractals. Note that the PowerMac native version is only available to registered users. PowerMac owners using unregistered copies of CFG can get a good taste of the speed of the PowerPC code by running *powerCFG* 2.21, which generates four

fractal types at very high speeds.

### **New in Version 2.3**

The major feature of version 2.3 is the addition of ten new fractal types. All types are similar to the Mandelbrot - Julia fractals. A unique color is generated for each pixel based on a complex algorithm. The algorithms reside in two classes: transcendental and polynomial. Both classes are MUCH more processor intensive than the traditional Mandelbrot type.

Additional changes in version 2.3 are increased PowerMac performance, reorganized fractal parameter dialog boxes, and making CFG a "fat binary" (one application contains both the PowerPC and 680x0 code). **The fat binary is included on the registered copy only** (see below).

On the negative side, support for versions of the system software earlier than 7.0 has been removed. Also, conversion of file formats previous to 2.3 has been moved to a separate utility program. These both contribute to increased ease of code maintenance and decreased code size.

Finally, the unregistered version does not contain any PowerMac code. This was done to provide incentive for registering the software. However, to show the extreme speed of the PowerPC, the PowerMac version of 2.21 is included with the unregistered version.

### **CFG Contents**

#### Unregistered Version

CFG 2.3	unregistered version (68K code only)
powerCFG 2.21	unregistered version of CFG (to demonstrate awesome speed of PowerPC native product)
Registration Form	form for registering CFG

miniManual	short program overview, registration information, and information on obtaining the manuals in non-Microsoft Word format
Settings <i>f</i>	sample fractal settings
Palettes <i>f</i>	C source code for creating your own palettes (THINK C 6.0 format) and some sample palettes
CFG Manual	full CFG manual (this manual)
CFG Scripting Guide	technical documentation of scripting CFG
AppleScript <i>f</i>	sample AppleScript files
CFG & Frontier <i>f</i>	UserLand Frontier install files and DocServer documentation for scripting
movieDemo	sample Frontier script (and two support scripts) that creates a PICS fractal movie
cfgMovieDemo.ucmd	"plug-in" for Frontier to add functionality required by movieDemo; includes C source code for the UCMD
μCinema Convert	simple PICS to QuickTime conversion application and C source code

### Standard Version

Everything from "Unregistered Version" with the following exceptions:  
a fat binary of the registered version of CFG 2.3 is included instead of CFG 2.3 and native CFG 2.21.

Master Version

Everything from "Standard Version" plus:

Source <i>f</i>	full source code to CFG in Metrowerks CodeWarrior format (only slight modifications needed to compile under THINK C 6.0 or later); includes portions of Frontier SDK 3.0 which are used to build CFG
-----------------	--

**Very Brief Background**

The original fractals in this program (Mandelbrot, Julia, and RandomWalk) are the result of pseudo code taken from *Scientific American*. Other fractal algorithms were developed from the following books: *Fractals for the Macintosh*, *Fractal Programming in C*, and *Advanced Fractal Programming in C*. Note that although these books provide a good basis for developing the fractal algorithm, they are by no means complete. Many times code samples were simply incorrect or additional (unprovided) information was required to create a functional algorithm [whether or not this was intentional...]. Special thanks is extended to *Advanced Engineering Mathematics* (by Erwin Kreyszig) and its coverage of complex analysis for its help in finding the missing pieces.

The Mandelbrot and Julia fractals are the fractals that most people have seen previously. The Dragon fractal family is very similar to both Mandelbrot and Julia families. In these fractals, the screen is a complex coordinate grid. There is one constant complex number and one that varies. Each pixel is assigned a color by an algorithm that determines the amount of iterations needed for the algorithm to reach an escape ("freedom") value. A limit is placed on the iterations ("accuracy")

which stops the

calculation if it seems that the algorithm will never reach the freedom value. The lower these freedom and accuracy values are set, the faster the fractals can be generated.

Colors are determined by the number of iterations required to reach the freedom value. If a pixel does not reach freedom, it is colored black. If a screen location is already beyond the freedom value, the pixel is colored white.

The transcendental and polynomial fractal classes are similar to the Mandelbrot class in the general fractal creation method. The transcendental algorithm uses the complex version of sine, cosine, and similar functions. Polynomial algorithms use complex polynomial equations of an order  $N$  (where  $N$  is greater than 1). In fact, the Mandelbrot set is just a type of polynomial algorithm.

The Random Walk algorithm places a pixel in the center of the screen. It then places one at the edge of an invisible circle and the computer moves the pixel in random directions. It stops when the pixel touches a colored pixel or goes outside the circle. This process continues until some number of colored pixels are displayed on the screen. An option exists to display the point as it wanders aimlessly. Note: This algorithm is quite time consuming. For maximum speed, do not display the walk (which is not visibly on very fast systems anyway), just display the end result. The walk is entertaining, but only for a few minutes (not a few hours or days).

The circle size for Random Walk constantly changes in an attempt to limit the time required for fractal generation. This circle is not too close to the central colored pixels, but not so far that the generation time becomes ridiculous.

### **Detailed Fractal Description**

Below is a more in depth description of the fractals and their algorithms.

The general algorithm, coloring process, and equations related to the fractal are shown.



## Mandelbrot Set

The Mandelbrot set is the fractal type which is most well known. The fractal is a color representation of an equation composed of complex numbers. The Mandelbrot equation is:

$$Z_{i+1} = z_i^2 + k$$

where  $z$  is a complex variable and  $k$  is a complex constant. The equation is iterative. One determines the value of the next iteration (variable with  $i+1$  subscript) based on the current value (variable with  $i$  subscript) of the variable.

Different screen pixels represent different complex variables. In fact, CFG uses the screen as a complex plane. The horizontal axis of this plane represents the real component of the variable, while the vertical axis contains the imaginary part. Moving from left to right increases the real component, while moving from bottom to top increases the imaginary component of the complex number. The value of each pixel is obtained by combining the real and imaginary components to produce the initial complex number used in the fractal calculations.

The coloring of a pixel is directly related to the iteration process. It is desirable to find out whether the equation is well behaved and approaches some limiting value or if the value will forever increase. Since time is a consideration, the equation cannot be executed forever. Constraints are placed on the equation. The equation must reach a "freedom" value within a given number of iterations to be considered to be increasing infinitely. If the equation does not reach the freedom value within the number of iterations, the pixel is set to black. If the value of the pixel is beyond the freedom value before any iterations, it is assigned to white. Otherwise, the color depends upon how quickly the fractal point reached the freedom value.

By modifying the freedom value and number of iterations, the user can change the color of the fractal. CFG implements both of these values as natural

numbers (not floating point numbers). These numbers can be modified by selecting *Mandelbrot* from the hierarchical menu *Parameters* which is under the *Fractals* menu. Colors are also altered by changing the scale at which the fractal is viewed as well as the fractal center and the constant number.

### **Julia Set**

The Julia set is related to the Mandelbrot set. In fact, the same equation is used. The difference is that instead of examining the value of the equation by varying the variable ( $z$ ),  $z$  is held constant while  $k$  varies. The effect is to look at a single point in the Mandelbrot set and see its value when using different constants. The coloring scheme remains the same.

Please note that the parameters dialog box has a slightly different meaning for the Julia set than for the Mandelbrot fractal. Center Real and Center Imag are the locations in the Mandelbrot set and form the fixed reference point (constant). Const Real and Const Imag form the center of the "constant", which is actually a variable in the Julia set.

### **Dragon Curves**

The Dragon curves are similar to Mandelbrot and Julia sets since the Dragon curves use the complex coordinate plane with a variable and constant (which is held fixed). The equation for these curves is:

$$z_{i+1} = kz_i(1-z_i)$$

Again,  $z$  is the complex number and  $k$  is the constant. The coloring method is also the same as the Mandelbrot and Julia sets.

### Polynomial Class

Equations of this class are similar to the Mandelbrot set (and the Mandelbrot set is just a subset of this class). The general equation for this fractal type are:

$$z_{i+1} = z_i^n + z_i^{n-1} + \dots + z_i^2 + z_i^1 + k$$

where  $n$  is a value greater than 1. Again,  $z$  is the complex number and  $k$  is a complex constant.

Version 2.3 adds the following five polynomial fractals to Color Fractal Generator: Tchebychev T ( $n=5$ ), Tchebychev C ( $n=6$ ), Legendre ( $n=4$ ), Laguerre ( $n=4$ ), and Hermite ( $n=4$ ). Definitions of these algorithms exist for  $n=2$  through  $n=7$ . Future versions of CFG will incorporate more (or all) of these fractal types.

Listed below are the iterative equations for the polynomial fractals:

Tchebychev T5:  $z_{i+1} = k (16 z_i^5 - 20 z_i^3 + 5 z_i)$

Tchebychev C6:  $z_{i+1} = k (z_i^6 - 6 z_i^4 + 9 z_i^2 - 2)$

Legendre 4:  $z_{i+1} = (35 z_i^6 - 30 z_i^2 + 3) / 8 + k$

Laguerre 4:  $z_{i+1} = k (z_i^4 - 16 z_i^3 + 72 z_i^2 - 96 z_i - 24) / 24$

Hermite 4:  $z_{i+1} = k (16 z_i^4 - 28 z_i^2 + 12)$

### Transcendental Class

The generation method for these equations is similar to the above described sets. The code iterates through the equation until an escape value or iteration limit is reached. It differs in that it does not use a polynomial equation, but uses complex versions of the transcendental functions. A multiplier exists in addition to a constant. Below is a equation for cosine:

$$z_{i+1} = m \cos (z) + k$$

where  $z$  is the complex number,  $k$  is a complex constant, and  $m$  is a complex multiplier.

The fractals of this class that CFG supports are: cosine, sine, exponential, hyperbolic cosine, and hyperbolic sine.

### **Random Walk Fractals**

The final fractal type present in Color Fractal Generator is the Random Walk fractal. This fractal is completely different from the previous three types. This fractal type lies on a simple integer plane, with horizontal values increasing left to right and vertical value increasing bottom to top. The plane size is limited to the size of the fractal (set by the 'Size' menu).

A single pixel (colored black) is set in the center of the plane. A new point is placed on an invisible circle whose center is the initial pixel. The computer moves the point in random directions. This process ceases when any of the following occur: the point moves outside this circle, the maximum number of steps have elapsed, or the point touches any colored pixel (initially, only the single black pixel is considered colored). If the point moves outside the circle or the maximum number of steps elapse, the point is discarded and a new one starts this process. If the point remains (it touched another colored pixel), the location of the point is colored according to user defined preferences. This process continues until some number of points is displayed. This number is also user defined.

There are six coloring schemes used by CFG: direction that the point last traveled, direction that the fractal touched, the angle formed by drawing a line from the fractal center to the new pixel, the time that the point was "walking", the distance from the pixel to the fractal center, and the distance between the current point and the last point that CFG placed. As many options as desired may be set, but at least one of these options must be set. The *Generating Fractals* section provides more insight into the meaning of the color options. If more than one

option is used, the resulting color is an average of colors that would have resulted from each of the coloring options individually.

## **Generating Fractals**

There are two methods to initiate a fractal display: create a new fractal or open a saved fractal. To create a new fractal, simply select *New* from the *File* menu. Opening a saved fractal is just as easy: choose *Open* from the *File* menu and select the saved file from the standard open dialog box. These methods are OK for either opening pre-saved fractals or seeing the default fractal, but to get the most out of this program, the user will need (and almost certainly desire) to do more.

## **Basic Concepts**

Several concepts need to be understood before undertaking the fractal creation process. Basic concepts include the *New Parameters* window, image size, and generation speed. The most important of these ideas is the *New Parameters* window (see Figure 1), which contains information about the fractals that will be created.

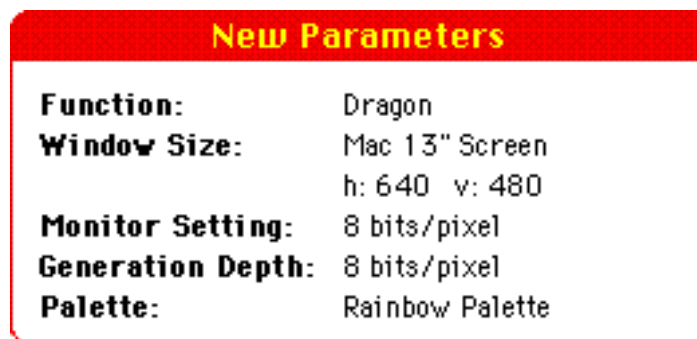


Figure 1: *New Parameters* Window

The *New Parameters* window displays the current function, window size, monitor setting, fractal creation depth, and the palette which will be applied to new fractals. Many of these values can be modified by different menu options. The function is the fractal that will be generated when *New* is selected. This item can be modified by selecting a fractal from the *Function* hierarchical sub-menu of the *Fractal* menu. The window size determines the image size of the fractal. The first line is the name corresponding to the *Size* menu, while second row contains the image's horizontal and vertical size in pixels. The Monitor Setting is the color depth of the monitor with the most colors in your system. Generation Depth is the depth at which the image will be created. This value will be the same as Monitor Setting unless a specific depth is selected from the Preferences dialog box (see the *Preferences* section). Finally, Palette is the name of the palette which will be applied to new fractals.

Changes to items listed in the *New Parameters* window will be displayed when they are made. However, these changes must be made with either the *Help* window or *New Parameters* window as the front window. Changes made with other windows in the front apply to those windows individually and have no effect on the "global" parameters. This is very important. The global parameters are what control the creation of new parameters while changing parameters with a fractal window as the front window changes the window's "local" parameters. This permits modification of a currently open fractal without changing parameters of fractals that will be created. This also works in reverse. One may change the global parameters for new fractals without modifying currently open fractals.

The size of a fractal is completely up to the user. The menu lists several standard image sizes. A custom size may also be selected. Information on each of these sizes is located in the *Menus* section (under the *Size* menu) and in the *Fractal Size and Memory* section.

It is also important to choose the speed at which to generate fractals. First decide whether to generate the fractal in the foreground or background. Foreground processing is the fastest method, but only one fractal may be generated at a time. More importantly, the Macintosh cannot be used for other tasks while processing in this mode. Background processing offers the advantages of being able to perform other duties while creating fractals. One may choose from three background speeds: slow, medium and fast. As expected, slow generates a fractal slowly, providing the most time for other applications, while fast creates a fractal more quickly, but uses more processing power. In most cases, run in medium or fast mode. Only operate in slow mode if you are performing other time consuming tasks that require lots of processing power (or if you have a machine without a math coprocessor).

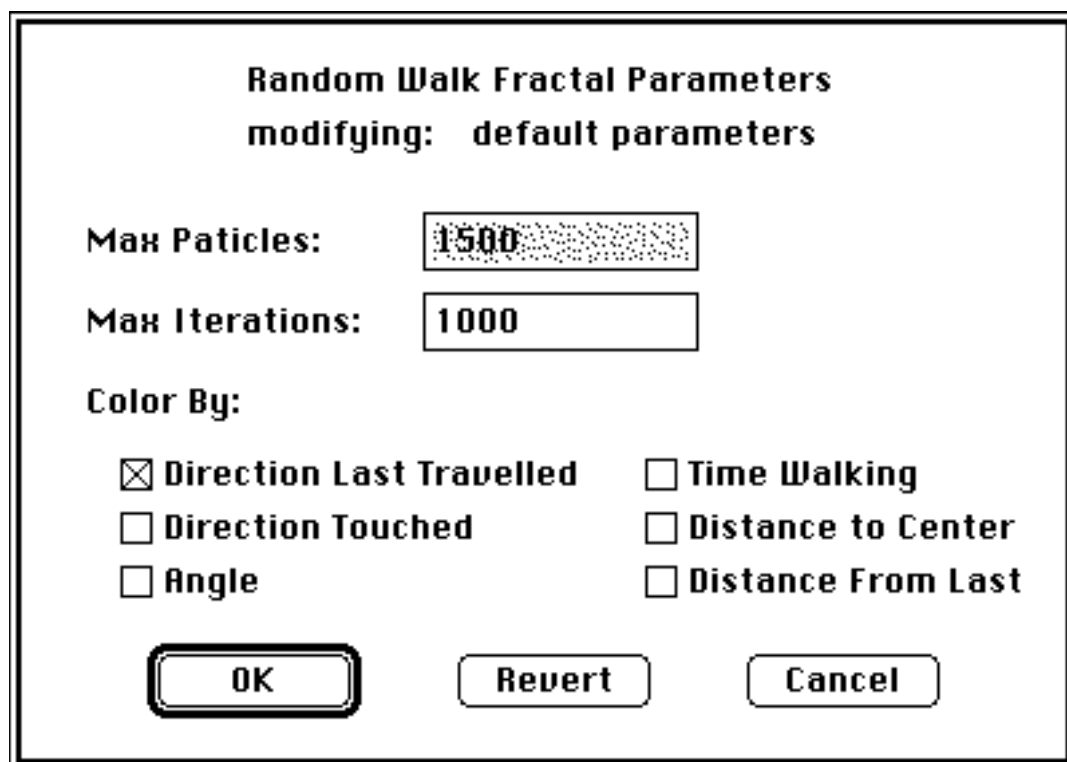
To stop the fractal creation process for foreground generation, press the command (apple) and period keys simultaneously or if the progress scale is displayed (see *Preferences* section), press the cancel button. For the background, press either the command (apple) and period keys simultaneously or select the *Stop* item from the *Fractal* menu. After stopping the generation, one can continue from where the creation process left off by selecting *Continue* from the *Fractal* menu. Selecting *Redraw* from this menu will clear the current fractal image and start generation over again.

While the fractal is generating in the background, many options will not be available. The palette cannot be accessed, a selection cannot be made, parameters cannot be changed, the image size stays fixed, and one may not zoom the image.

### **Fractal Parameters**

Each fractal has certain parameters associated with them which regulate how the fractal is created. Select the fractal window whose parameter you want to

modify or select either the *New Parameters* window or *Help* window to modify the global parameters (for future fractals). Select the fractal that you wish to modify from the *Parameters* sub-menu, which resides in the *Fractal* menu. If a fractal window is the active window, only the parameters that apply to the window's fractal function may be modified. If the fractal was a Random Walk fractal, a dialog box similar to Figure 2 is displayed. Figures 3, 4, and 5 show sample Classic (Mandelbrot, Julia, Dragon), Transcendental, and Polynomial dialog boxes, respectively.



Figure

2: Random Walk Parameters Dialog Box

The parameters applicable to Random Walk fractals determine the maximum number of particles to display, how long to wait for contact, and the coloring method for each pixel. The Maximum Particles item sets the maximum number of pixels to be colorized using the Random Walk algorithm. A 640 x 480



pixel screen has over 300,000 pixels. Coloring all of these by the Random Walk method would take months to generate (if at all possible since the method leaves gaps in the fractal). A good number might be between 1000 and 5000 points. However, be prepared to wait a while. Max Iterations sets the number of steps to make before discarding the point. If the number is too low, the roaming points may never reach other pixels. If it is too high, time might be wasted while the point wanders in a region with few pixels to impact. Generally, it is good to err on the large size. Note: The default setting is 1000 iterations. This might be quite low for large walks.

There are six coloring options for Random Walk fractals: direction that the point last traveled, direction that the fractal touched, the angle formed by drawing a line from the fractal center to the new pixel, the time that the point was "walking", the distance from the pixel to the fractal center, and the distance between the current point and the last point that CFG placed. One may select any combination of these coloring schemes. If more than one method is used, the colors produced by each option are averaged to produce the pixel color.

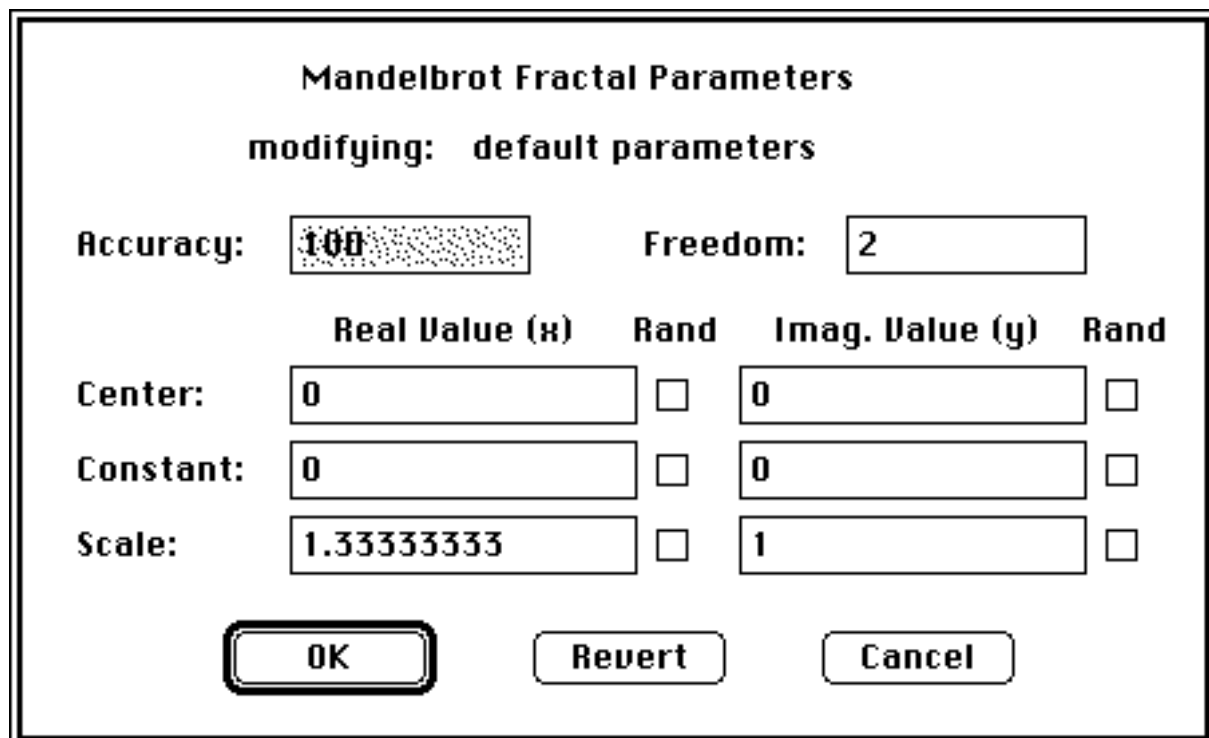
Direction that the point last traveled permits up to four colors for a pixel: one for left, right, up, and down. Similarly, direction that the fractal touched a pixel provides only four colors.

The angle formed by drawing a line from the fractal center to the new pixel permits many fractal colors and will produce a radial color pattern since the color depends upon the angle. This coloring method uses floating point numbers to generate the angle and can produce as many colors as can be displayed.

The time that the point was "walking" produces a color based upon the number of steps required to collide with the point. This method produces at most Max Iterations colors.

The two distance coloring methods calculate the distance between two points. The maximum distance would be the distance between two corners that lie on the same diagonal. The distance from the pixel to the fractal center method will produce concentric coloring of fractals.

Notice that below the dialog box title is the following text: "modifying: default parameters". This informs the user that the global parameters are being modified. If a fractal window's parameters were being altered, the window name would appear after the "modifying:" message (see Figure 3 for an example).



**Mandelbrot Fractal Parameters**

**modifying: default parameters**

**Accuracy:**  **Freedom:**

	Real Value (x)	Rand	Imag. Value (y)	Rand
<b>Center:</b>	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
<b>Constant:</b>	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
<b>Scale:</b>	<input type="text" value="1.33333333"/>	<input type="checkbox"/>	<input type="text" value="1"/>	<input type="checkbox"/>

**OK** **Revert** **Cancel**

Fig

Figure 3: General Parameters Dialog Box

The Mandelbrot, Julia, and Dragon Parameters Dialog will be referred to as the General Parameters Dialog. There are two types of data that can be modified in the General Parameters Dialog: the actual variables or a flag indicating random creation. The variable text boxes will contain the current parameters. Modify these to select an exact location to examine. It is better to leave these to the scaling

routines. The only exact parameters not affected by zoom is the constant (Const for Mandelbrot and Dragon, Center for Julia). These values should be modified by the user as no other method will alter their contents.

Setting any of the random check boxes forces the variable to be randomly generated on the next Redraw. Since the method generates a random number within a fractal specific range, there is a good chance that the value may be of little aesthetic interest once the fractal is generated. To help guard against this, select the *Limit Extreme* item from the *Special* menu before redrawing. This forces random variable generation within a more confined range. Note that the random flag is cleared after the fractal is generated to permit zooming. Otherwise, on a zoom the variable would be generated at random again, effectively throwing the user into an unknown and unwanted location.

Notice that below the dialog box title is the following text: "modifying: Dragon 1". This informs the user that a fractal window's parameters are being modified. If global parameters were being altered, the following text would appear: "modifying: default parameters" (see Figure 2 for an example).

Figure 4 and 5 show the fractal parameter dialog boxes for transcendental and polynomial fractals, respectively. These dialog boxes are quite similar to the General Parameters dialog (see Figure 3 above). The "Iterations" and "Break Value" fields map to the General Parameter's "Accuracy" and "Freedom" Fields. The transcendental fractals have an additional set of edit text boxes for entry of the multiplier. At each iteration, the fractal value is multiplied by this value before adding the screen location constant. Transcendental fractals also permit randomly generated break values.

**Cosine Fractal Parameters**

**modifying: default parameters**

Iterations:  Break Value:  ☐

	Real Value (x)	Rand	Imag. Value (y)	Rand
Center:	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Constant:	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Scale:	<input type="text" value="2.0943951"/>	<input type="checkbox"/>	<input type="text" value="1.57079633"/>	<input type="checkbox"/>
Multiplier:	<input type="text" value="1"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>

Figure 4: Transcendental Class Parameters Dialog Box

**Tchebychev T Fractal**

**modifying: default parameters**

Iterations:  Break Value:

	Real Value (x)	Rand	Imag. Value (y)	Rand
Center:	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Constant:	<input type="text" value="0.654"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Scale:	<input type="text" value="1.33333333"/>	<input type="checkbox"/>	<input type="text" value="1"/>	<input type="checkbox"/>

Figure 5: Polynomial Class Parameters Dialog Box

## Selection

Any portion of the fractal may be selected by clicking the mouse on the desired start location, holding the mouse button down, and dragging to select a rectangular region. A selection may be enlarged by holding down the shift button and clicking the mouse button. The bottom right corner will now be at the location of the mouse. If you hold down the option button, the selected area will be proportional to the image rectangle. Color Fractal Generator only supports single rectangular selections. The menu command *Select All* will select the entire image. A selection may not be made on a fractal that is being created.

Once an area is selected, several actions may be performed on it. The user may zoom in on the area; invert the color in the region; and cut, copy, paste, or clear the selection. Zooming is covered in the subsequent section. Coverage of the Edit functions (cut, copy, paste, and clear) is found in the *Menus* section under the Edit menu.

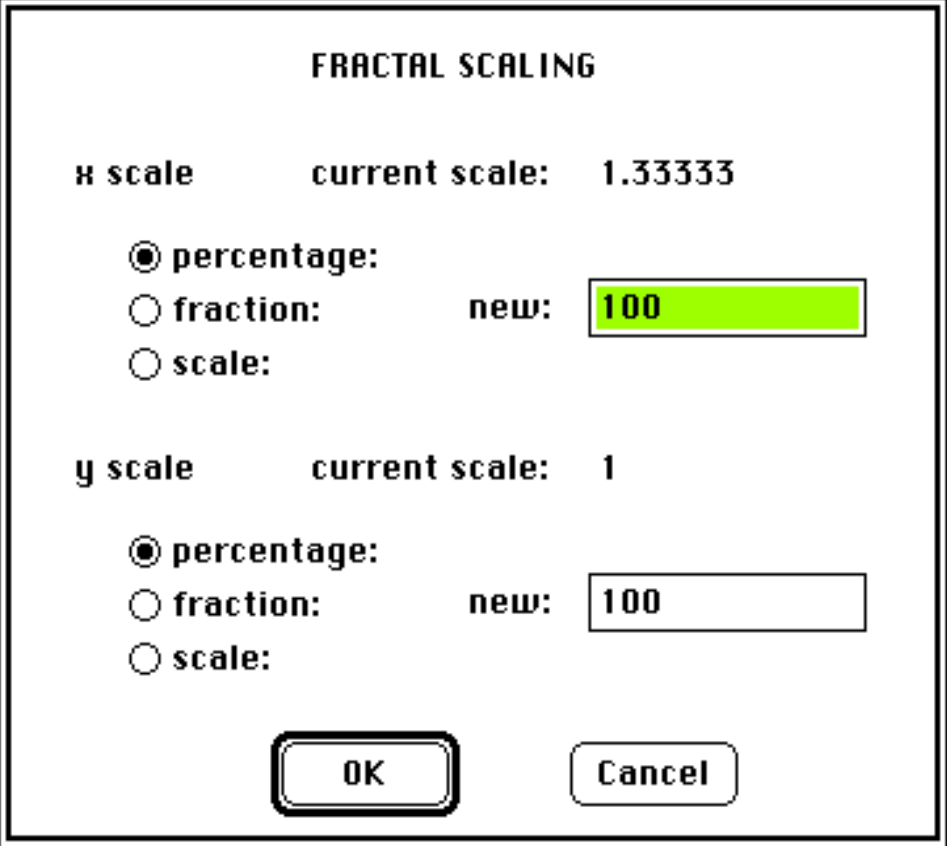
Inverting the color of a selection will reverse the colors in the area. The exact inversion method depends on the color environment (see *Applying Color* for information of the color environment). If the color is direct color, the data bits of the color information are flipped. White becomes black, black becomes white, pure red would lose all red information and be a blend of green and blue, etc. This is a true inversion. For indexed color, black and white are flipped. But for other colors, their indexes are swapped. No new colors are created. The existing colors are simply moved around in the image.

## Zoom

There are three zoom commands which result in four zoom actions. None of these commands function on a Random Walk fractal (since there is nothing to zoom in on). Under the *Special* menu are the commands *Zoom In*, *Zoom Out*, and *Zoom*. *Zoom In* retains the current fractal center, but halves the scales. This is a 2X zoom. *Zoom out* is a 0.5X zoom. The scales are doubled. The only limit to the zoom amount is the precision of the machine.

If a rectangle of the fractal is selected, the *Zoom* take the selection and stretches it to fit the image size. The center of the selection becomes the fractal center, while the edges of the selection become the edges of the fractal image. It is possible that the real and imaginary portions of the image will be scaled by different amounts, since the image aspect ratio is not maintained in the selection. To scale both real and imaginary data the same amount, hold down the option key when making the selection. This forces the selection to maintain the image aspect ratio.

If nothing is selected when *Zoom* is chosen, a dialog box appears prompting the user for the scaling information (see Figure 6). This dialog box permits precise control of zooming.



**FRACTAL SCALING**

**x scale**      **current scale:**    1.33333

☒ **percentage:**  
☐ **fraction:**      **new:** 100  
☐ **scale:**

**y scale**      **current scale:**    1

☒ **percentage:**  
☐ **fraction:**      **new:** 100  
☐ **scale:**

OK
Cancel

Figure 6: Fractal Scaling Dialog

The user may modify the x and y scale separately. The current value is displayed in text boxes next to the text: "current scale:". The scaling method for each scale is selected by the two sets of radio controls. Select percentage if you want to enter the scale by percent, fraction to enter a multiplier, or scale to enter the exact scale value.

The percentage is how much you want to zoom in. Entering 200 will zoom in 2X (200%). The fraction is a multiplier to zoom in. Entering 2 with a fraction zooms to 2X. Numbers less than or equal to 0 are invalid.

## **Applying Color**

The main attraction of fractal programs is color. This program tries to permit a wide use of color and application of color to the fractal. Color Fractal Generator can handle anything from 16 to 16.7 million colors. Varying the palette and fractal parameters results in different color application to image. A general description of the way altering parameters affects fractals is discussed in the *Generating Fractals* section. Below is a detailed explanation of the different color depths and use of palettes to alter a fractal's appearance.

The Macintosh determines the colors of each pixels in an image by examining the pixel's value. The two techniques used for converting a pixel value into a color are a color look-up table (CLUT) (indirect color) or the pixel's value as the actual color (direct color). If a monitor is set to 16 or 256 colors (4 or 8 bits of color per pixel), then the Macintosh uses a CLUT. If the monitor is set to more colors (or bits) per pixel, the image is direct color.

Indirect colors require less memory to display an image, but only a limited number of colors can be displayed at once. The value of each pixel in this type of image represents an index into the CLUT. Each index has a corresponding color associated with it. By referencing this table, one converts from the pixel value to a color. Also, changing entries in the CLUT changes the color (since the index will reference a different color). When the user selects or loads a palette, Color Fractal Generator build a CLUT with colors that would be present in such a palette. For example, the majority of entries in a rainbow palette consist of very bright colors that span the color spectrum (from red to yellow to green to blue), while the entries in a grayscale palette are shades of gray.

The user can specify how many entries in the palette to use, see Figure 7. The minimum amount is four colors, while the maximum is either 10 (for a 16 color monitor) or 250 (for a 256 color monitor). The reason that there are some



entries that are not used is to allow for other colors required by the system, such as white, black, and grays. The

maximum number of colors to use is set by the user by the *Max Colors* command of the *Special* menu. Note that altering the number of colors will either truncate or expand the current palette. This may have some unwanted side effects in fractal coloring.



Figure 7: Maximum Colors Dialog

A direct color image is quite different. Instead of using a color look-up table, the pixel values contain color information required by the monitor. Colors are generated by applying different intensities of three primary colors: red, green, and blue. Each pixel contains a certain number of bits for each of these colors to represent the intensity. With more bits per pixel, more colors can be represented. Therefore, 32 bits per pixel (24 of which are color information) can represent millions more colors than 16 bits per pixel (15 bits of which have color data).

Color Fractal Generator uses the advantages of direct color data to greatly increase the number of colors displayed in a fractal. The palette is used as a guideline for selecting image colors. To maintain the highest resolution, the palette in a direct color environment is always set to 250 colors. As a fractal is created, CFG uses the fractal data to determine which two colors in the palette to use as color guides. CFG uses linear interpolation to select the exact color (which lies somewhere between the two reference colors). However,

for the rainbow and gray scale palettes, an exact algorithm is used to determine the actual color.

Whether using indirect or direct color, using palettes is necessary. The standard palettes are displayed in a hierarchical menu found at the *Standard Palettes* item of the *Edit* menu. All items above the separator line are default palettes which are always present in the program. One item will be displayed below the line. It is either *No Palette Loaded* or the name of a custom or loaded palette. Initially, this item will read as No Palette Loaded. However, if at any time a palette is edited or loaded, the name of the last custom palette will always appear as the final menu entry.

The custom palette listed for each fractal window could be different. Be aware that the Help window and *New Parameters* window display the global custom palette which is applied to new windows (if this palette is selected). These are important concepts. Editing or loading a palette for one window applies only to that window, not to any other fractal. Thus, you can modify the palette of any window without affecting the colors applied to another window. Also, you can select the palette for fractals before that fractal is opened or created.

A palette is applied to the desired window by bringing the window to the front (clicking on it) and selecting the desired palette from the hierarchical menu (*Standard Palettes* item of the *Edit* menu). If the window is either the *Help* window or *New Parameters* window, CFG applies the palette change to the global data (affecting fractals that are created in the future).

If an indirect color system is used, changing the palette may change the fractal image since the image data only contains references into the palette, not actual color information. For the direct color system, no image change will be evident until the fractal is regenerated as the colors are stored in the image data and the palette only acts as a guideline for selecting colors. To guarantee the color change, after altering the palette, regenerate the fractal.

It is possible that none of the palettes are to your liking (only three palettes are built into CFG). To create an new palette or alter an existing one, select the maximum desired colors (by using the *Max Colors* command from the *Special* menu, again this only works for indirect color). Select the palette which you wish to modify. Now select *Edit Palette* from the *Edit* menu. A dialog box with the palette's colors appears (an example is shown in Figure 8; the colors will not appear as good as possible unless the monitor is set to more than 8 bits per pixel).

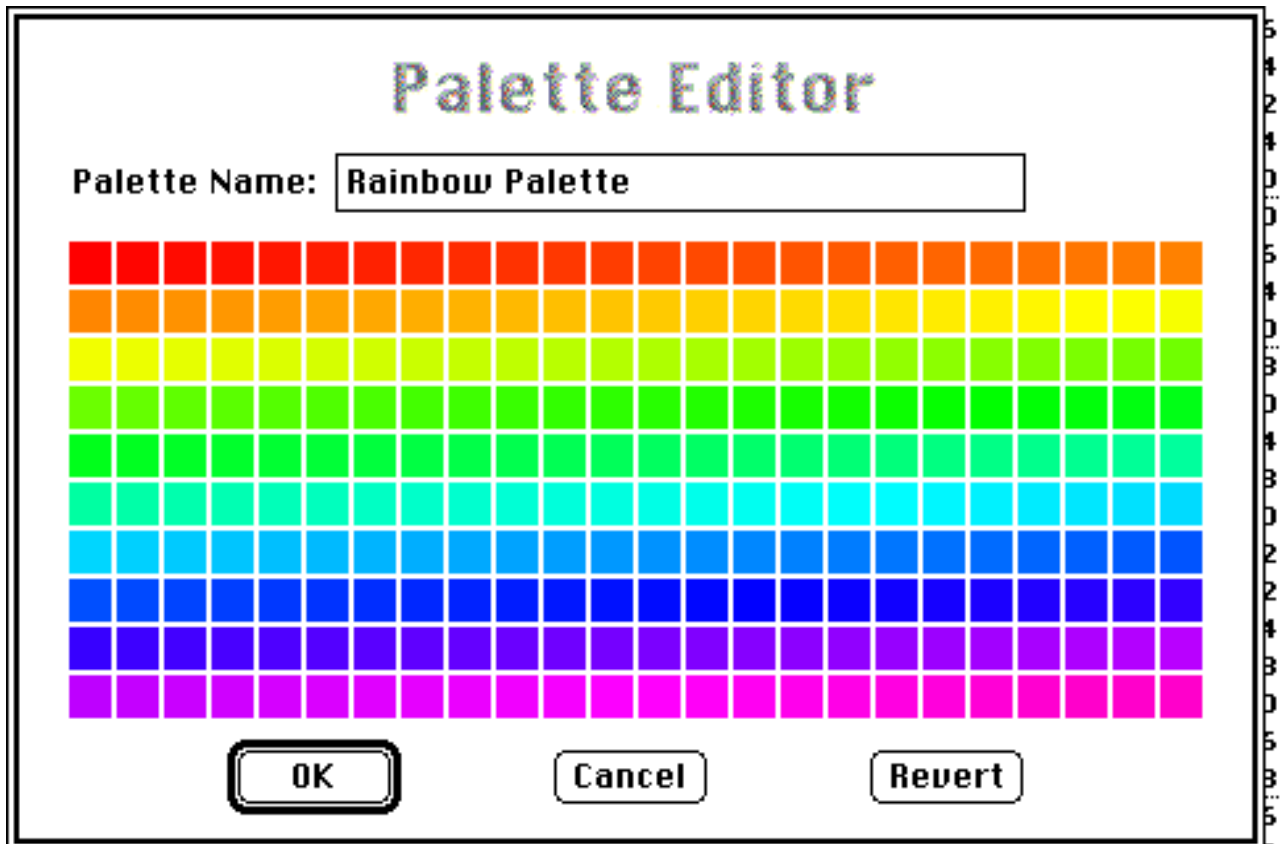


Figure 8: Palette Editor

The user may modify the palette name by using the editable text box. You can select a color by clicking on the color with the mouse or hitting tab to move from the text field to the color field and using arrow keys to maneuver to the desired color. To edit a color, either double click on it, or simultaneously press the Command (Apple) and Return

keys. Be careful not to just strike Return since that closes the dialog box. When editing a color, the color picker is displayed which permits the user to select the custom color. After selecting the color, the color will be displayed in the appropriate box. If the colors do not appeal to you, just reset them to the same state as when you entered the Palette Editor by striking the Revert key. Clicking the OK key makes the modified palette a custom palette and applies it to the current window. Striking the Cancel button closes the Palette Editor without changing any colors.

Any palette may be saved by selecting *Save Palette* from the *Edit* menu. The currently active palette of the front-most window will be saved. The user just needs to respond to the standard save dialog to select the palette's file name and location. A palette may also be saved in the parameters file of a fractal. To do so, choose *Preferences* from the *File* menu and check the *Save Palette* box. Next, choose either *Save* or *Save As* from the *File* menu to update the current settings file or create a new one. The palette will be saved with the settings.

To open a saved palette, simple choose the fractal window that you want the palette change to affect (making it the front-most window). Select *Load Palette* from the *Edit* menu and choose the palette from the standard open dialog box (make sure the pop-up menu is set to CFG). The palette will be loaded as the custom palette and this palette will be selected. If the fractal uses indirect colors, the color change will be noted immediately. For direct color fractals, regenerate the image to apply the new colors. If the palette was saved in the settings file, opening this file will automatically apply the correct palette.

The user is no longer limited to only CFG created palettes. Now any palette located in a resource file can be used. Palette resource types include 'pltt' and 'clut'. To load a palette from a resource file, select *Load Palette* from the *Edit* menu. From the pop-up menu, choose either 'pltt' Resource or 'clut' Resource (see Figure 9). The list of files will now show all files. Choose the file with the resource type that you want to load. If the

specified resource type is not present in the file, and error message is displayed and no palette is loaded.

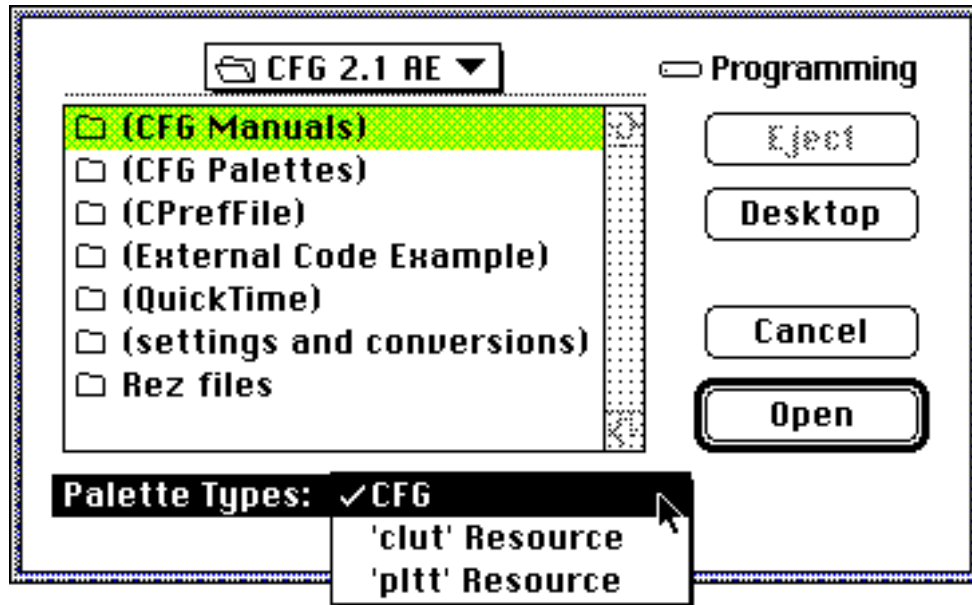


Figure 9: Load Palette Dialog

If only one resource of the specified type is found, it will be loaded as the new palette. If more than one resource of type 'pltt' or 'clut' is found, then a list of the resource IDs and names is shown in a dialog box (see Figure 10). Select a resource to load and strike the OK button, or choose the cancel button to skip the loading.

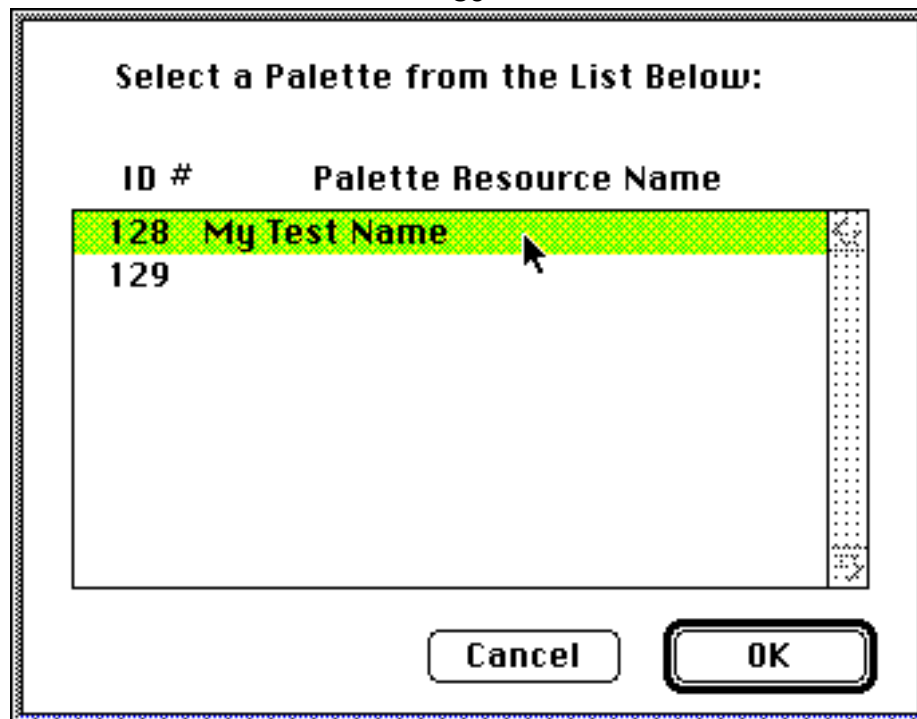


Figure 10: Resource Palette List Dialog

For 'pltt' resources, the first two and final four entries in the palette are ignored since these are reserved by CFG. For 'clut' resources, the first entry and the final five are ignored. The reserved entries are used by CFG for white, black, and a few other colors.

When loading a palette, the maximum number of colors is automatically adjusted for according to the number of entries in a palette. If only full palettes are used (all 256 colors defined - or 16 for 4 bit systems), then no problems should arise. However, loading a palette with a different number of colors on indirect systems may require a redraw before one can take advantage of the full palette. For example, if a fractal currently has a palette with 200 colors defined and a new palette is loaded with 250 colors, then one would need to redraw the image to view the newly defined 50 colors (the image only has data for the original 200 colors).

NOTE about changing screen depths: Try not change the screen depth while this program is running. Although not as hazardous as it would have been with previous

versions of CFG, it still may pose a danger. The safest method of changing the screen depth is to quit CFG, change the depth, and re-launch the program.

## **Saving Data**

There are four types of data that can be saved: data from the help window, palettes, parameters for fractal creation, and the fractal's graphic image. For information regarding saving the help information, see the section on the *Help System*. Saving palette information is described in the *Applying Color* section.

Data directly related to generating fractals can be saved and reloaded for future use. This enables you to continue exploration of a fractal where you left off. To save the fractal's settings, select either *Save* or *Save As* from the *File* menu. The difference between these menu items is that *Save As* will always prompt the user for a file name and location, while *Save* only presents this dialog once and continually saves the data to the same file. Saved data can be reopened with the *Open* command from the *File* menu.

The parameters file can contain additional information, such as the palette and fractal image (only for registered copies of Color Fractal Generator). Two preferences enable these options: *Save Fractal Image* and *Save Palette*. *Save Fractal Image* permits the user to save either a complete or partial fractal image inside the file. Thus, if generation could not be completed for some reason, the partial fractal image could be saved and one could start regenerating from where the creation process left off. Saving the palette inside a file prevents the user from needing to save the palette as a separate file and load the palette after loading the fractal.

To save the image as a PICT file, select *Save PICT* from the *File* menu. It will save either the complete or partial fractal image as a PICT file for opening in



another application. Only the actual picture is saved to this file, so one cannot continue exploration of the fractal world by only saving the image. The complete image is always saved unless

the window has a selection and the *Save Selection* preferences option is set. In this case, only the selection will be saved.

The *Save PICT* creates a color PICT file that many graphic applications can open. The user can modify the default program that opens the file by changing the file creator preferences (see the *Preferences* section below).

### **Opening Fractal Files**

CFG 2.3 can only open fractals created by version 2.3x. Code to open prior version was moved to another program called *Convert 2.3*. The main reason was to reduce code size and maintenance problems.

*Convert 2.3* is an application that accompanies CFG. It will convert data files from CFG 1.2 through CFG 2.2x to CFG 2.3 format. To convert the file, simply drag and drop the file onto the conversion application. Double clicking the conversion program has no effect. It only functions through the drag and drop interface.

### **Preferences**

Preferences permit the user to customize certain features in the program. Figure 11 displays the initial dialog box to expect after selecting the *Preferences* command from the *File* menu. This dialog contains three icon button to go to one of the three preferences sections. It also contains an OK button to keep any preferences changes and a Cancel button to discard preferences changes.

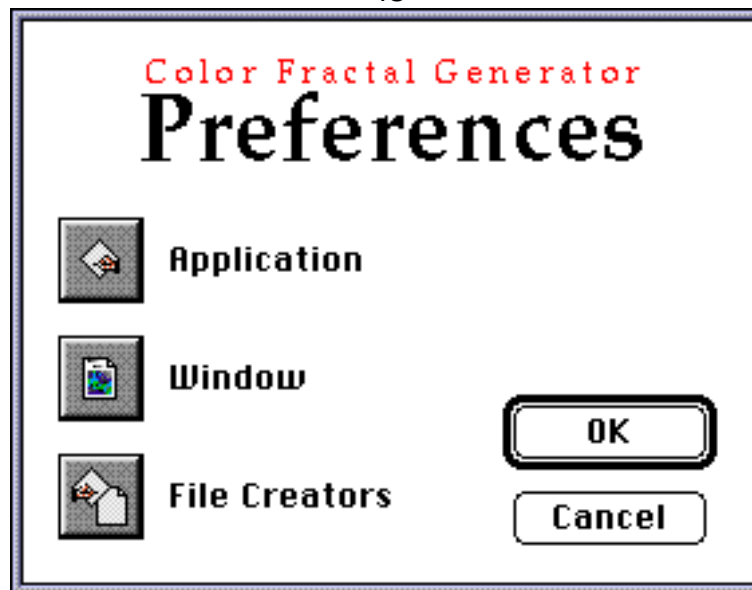


Figure 11: Preferences Dialog

As stated previously, preferences are divided into three sections: program, window, and file creator. The program section provides a set of options that apply to all fractal windows. Window preferences apply only to a single window. Every fractal may have a different set of window options. Finally, the file creator preferences provide a method to select the default application to launch for text and PICT files that are created by CFG.

### **Program Preferences Dialog**

The Program Preferences Dialog (displayed in Figure 12) is displayed in response to hitting the Application Icon. The items in this dialog box apply to all windows. This dialog controls the general user interface, including such features as numeric precision display, window updating, and progress scales. Each preferences item is described in detail below.

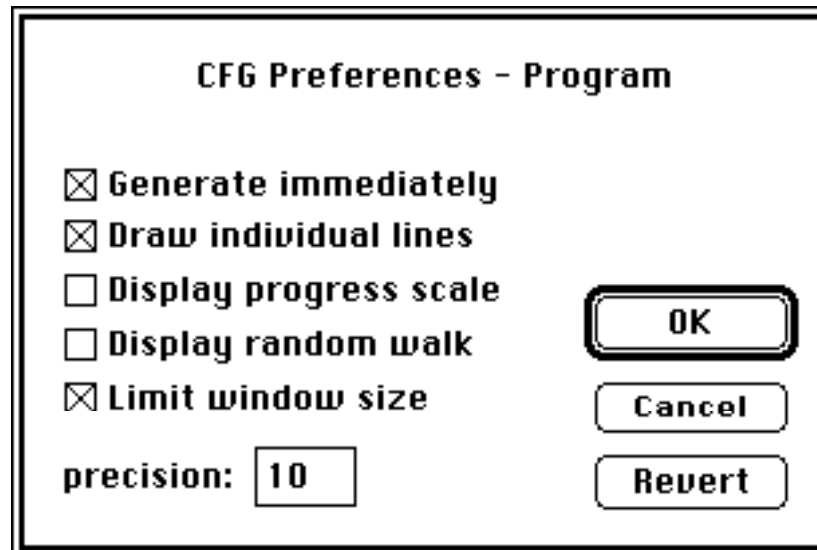


Figure 12: Program Preferences Dialog

#### Generate Immediately

Generate Immediately is an option which will start the image generation process as soon as a file is opened or a fractal is created (with the *New* command). The user does not have a chance to change any parameters associated with the window before fractal creation begins. If this option is not selected, no fractal creation will happen when *Open* or *New* is selected. Instead, an empty window will be displayed and it is up to the user to initiate fractal creation (with *Redraw*, *Continue*, *Zoom*, etc.).

#### Draw Individual Lines

Selecting this item will permit the application to update a fractal image as each line is completed, instead of presenting the entire fractal at once. This method is more lengthy, but it permits the user to see that the fractal is being created. If one desired to zoom in on a certain region, the user may stop the fractal creation once that region has been completed, long before the complete image would be created.

If this item is not checked, the image will only be displayed when the creation process is completed or has been canceled. Please note that this preference item does not apply to Random Walks, as these fractals are not created one line at a time.

#### Display Progress Scale

When checked, this option displays a progress scale while the fractal is being generated. As the fractal approaches completion, the scale is updated. An example of such a scale is shown in Figure 13. If the box is not checked, no scale is displayed.

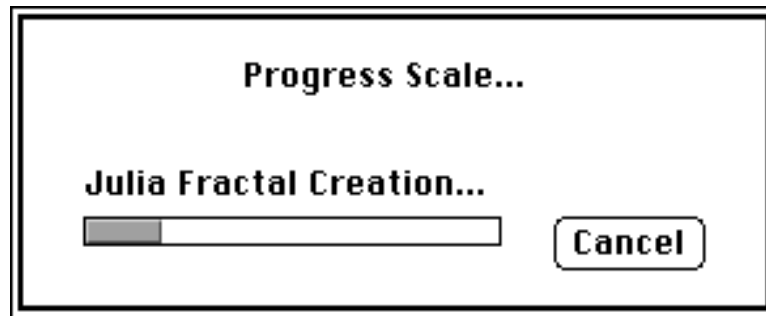


Figure 13: Progress Scale

This scale is only displayed while generating a fractal in the foreground. If a fractal is being created in the background, the scale will not be displayed, no matter how the preferences item is set.

#### Display Random Walk

This option applies only to Random Walk fractals and only when Foreground generation mode is active. If this option is selected, the actual movement of the particles is displayed as they create the Random Walk fractal. This method of fractal creation requires a great deal more time than without displaying the moving particles as the screen must be updated for each move. If the option is not selected or the fractal generation is in background mode, then no moving particles are shown.

#### Limit Window Size

This option prevents a window from being re-sized to larger than the fractal image. This is useful since it prevents a window from using excess screen area while displaying the entire fractal.

### Precision

This is the number of significant digits that are used to display fractal data. The value can be from 3 to 18 digits. The default is 9 digits.

### **Window Preferences Dialog**

The Window Preferences Dialog (displayed in Figure 14) is displayed in response to hitting the Window Icon. The items in this dialog box apply only to the top-most windows. If the top-most window is either the *Help* window or *New Parameters* window, then the preferences will be applied to newly created fractals. The controls in this dialog dictate the saving options and fractal depth. Each preferences item is described in detail below.

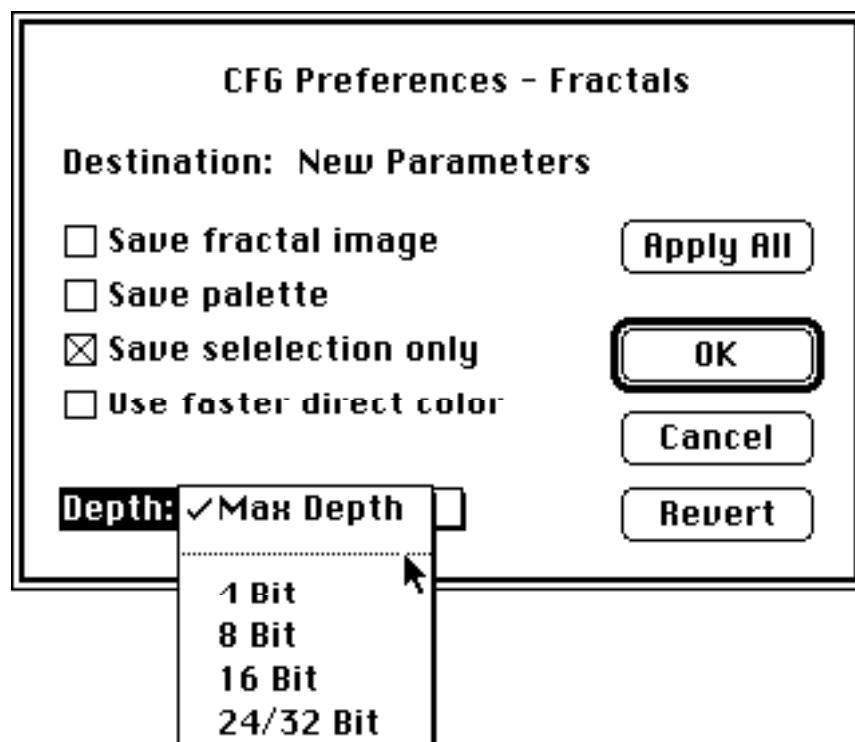


Figure 14: Window Preferences Dialog

### Destination

This is the window to which the preferences changes are applied. If the *New Parameters* window is specified, then changes apply the newly created fractals.

### Save Fractal Image

This item is only available to registered users. Checking this box will cause the complete or partial fractal image to be saved along with its settings. When the fractal is reopened, the image generation can pick up where it had left off. This option can create vary large files as it stores the actual fractal image in the file. If this option is not selected (as is the case for unregistered copies of this program), the fractal image will not be saved with the parameters.

### Save Palette

Setting this option saves the current palette (whether it is a standard palette or custom palette) in the settings file. This permits a custom palette to be applied to the fractal the next time it is opened without having to load the palette separately. Simply checking this box will not save the palette in the parameter file, however. The user must still select either *Save* or *Save As* from the *File* menu to update the file.

If this option is left unselected, the palette will not be saved to the parameter file. In this case, if the fractal had a custom palette associated with it, that palette will be lost when the file is opened from disk (unless it was saved with the Save Palette command). A default palette will be assigned to the fractal.



Save Selection Only

Checking this item permits saving only a selected region of the fractal. In response to the *Save PICT* menu option, if this item is checked and a selection is active in the fractal, only the selected region will be saved.

Use Faster Direct Color

Selecting this item causes CFG to use interpolation for direct color fractals using the Rainbow Palette instead of the standard system calls. This is especially useful for Power Macintosh computers since the standard method is very slow due (those portions of the Macintosh Toolbox are emulated). The fractal will be produced about three times faster, with only small color deviations from the standard coloring method.

Depth

The image's depth can be selected by the depth pop-up menu. The user may choose to create a fractal with as many colors as the most colorful monitor connected to the system or he may choose a specific color depth. A list of the menu choices and explanations is located in the *Pop-Up Menus* section under *Depth*.

One may choose to modify the depth of a currently open fractal. This can be dangerous since memory must be re-allocated. Make sure that the fractal is saved before altering its depth.

Apply All

This button will distribute the preferences to all open fractal windows. Thus any changes made to a particular window can easily be applied to all windows.

### Creator Preferences Dialog

The Creator Preferences Dialog (displayed in Figure 15) is displayed in response to hitting the Creator Icon. This dialog controls the application that will be assigned to opening any text and graphics files created by CFG. Each preferences item is described in detail below.

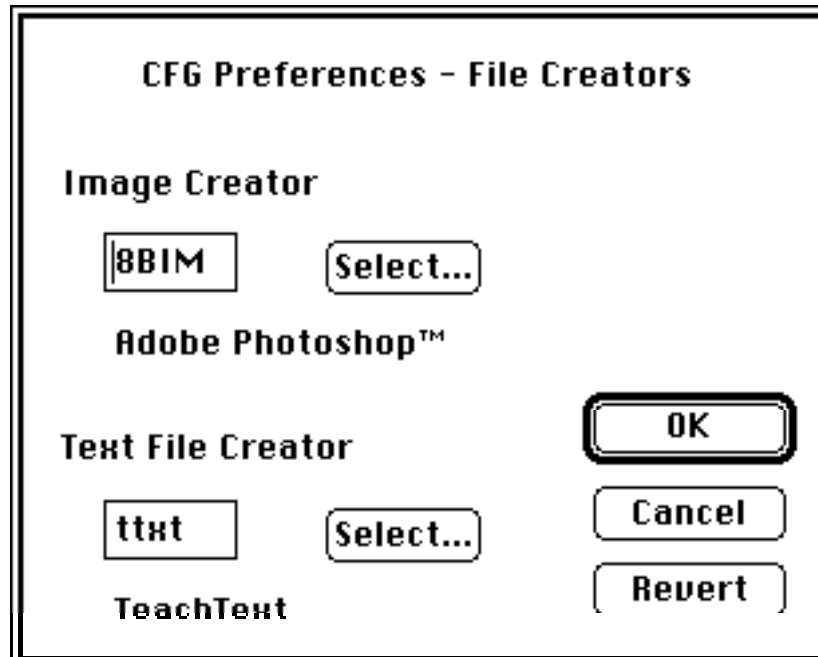


Figure 15: Creator Preferences Dialog

#### Image Creator

This set of controls permits the user to select an application as the owner of graphics files created by this program. One can either enter the 4 character signature into the editable text field or use the *Select* button to choose an application. An application chosen by the *Select* button will have its name displayed below the button and edit text field. If the signature is manually entered, no application name will appear.

### Text File Creator

This functions exactly as *Image Creator* above except that the creator applies to text files created by CFG.

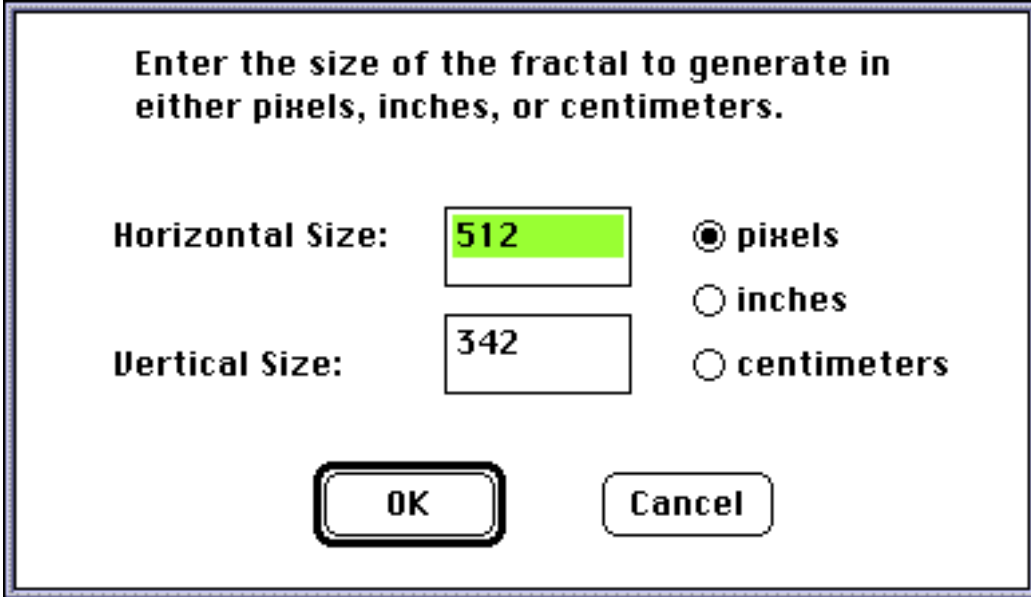
## **Fractal Size and Memory**

CFG can create fractals of different sizes. This not only permits custom use of the fractal, from using it for a background on a 19" monitor to creating an interesting mask for a QuickTime transition, but it permits the user to maximize speed and memory usage. Initially, the fractal generation size may be small so that the general fractal area can be found quickly (as generating a small image requires much less time than a large image). Once the desired fractal location has been found, the size can be increased for very detailed fractal image generation. Smaller fractal sizes also use less memory.

Size selection occurs through use of the Size menu. The basic selection include Screen Size, Mac 9" Screen, Mac 12" Screen, Mac 13" Screen, Mac 19" screen, and three QuickTime screen sizes. Screen Size sets the image size to the total amount of real estate used by all monitors connected to the system. Mac 9", 12", 13", and 19" Screen sizes set the image to the pixel size of those standard monitors. QuickTime sizes set the screen sizes to 160x120, 320x240, and 640x480.

The final menu item permits custom fractal size selection. Figure 16 displays the Custom Fractal Size dialog box. The user selects whether to specify the fractal size using either pixels, inches, or centimeters by toggling the appropriate radio buttons. The horizontal size and vertical size can be set by typing the sizes into the text boxes. There are minimum and maximum sizes for a window. The minimum size is 120 pixels horizontal and 100 vertical. The

maximum window size is 32000 x 32000 pixels (hope you have LOTS of RAM for this size).



**Enter the size of the fractal to generate in either pixels, inches, or centimeters.**

Horizontal Size:  ☒ pixels

Vertical Size:  ☐ inches

☐ centimeters

Figure 16: Custom Fractal Size Dialog Box

This program requires a large amount of memory, especially as the number of colors and image size increase. The settings of 1024K is fine for a monitor of size 640 x 480 pixels and an 8 bit video card. If you use a larger monitor or one with a video card with more depth (bits per pixel), the size of the memory allocated to the Color Fractal Generator should be increased. The base amount of memory to allocate to the program is 512K. For each desired fractal, add at least: bitmap size  $[(x \text{ pixels} * y \text{ pixels}) * \text{screen depth (bits/pixel)} / 8 / 1024]$  K. This is only a guideline. With some experimenting, the user will get a feel for the amount of memory needed for a desired image.

## **Help System**

Two help systems are provided in this program. The CFG Help Manager is accessed by using the *Help* command from the *Apple* menu. The other system is Balloon Help, which is only available to users with Balloon Help in their System (System 7.x and possibly late versions of System 6.0x). The CFG Help Manager is a subset of this

manual. Portions of the sections describing the operation of this program are included on-line. Balloon Help contains portions of this manual with additional information to provide help for systems not described in the CFG Help Manager (such as individual menu items and dialog components). The CFG Help Manager is described below. For information on operating Balloon Help, please consult the documentation which came with the System software.

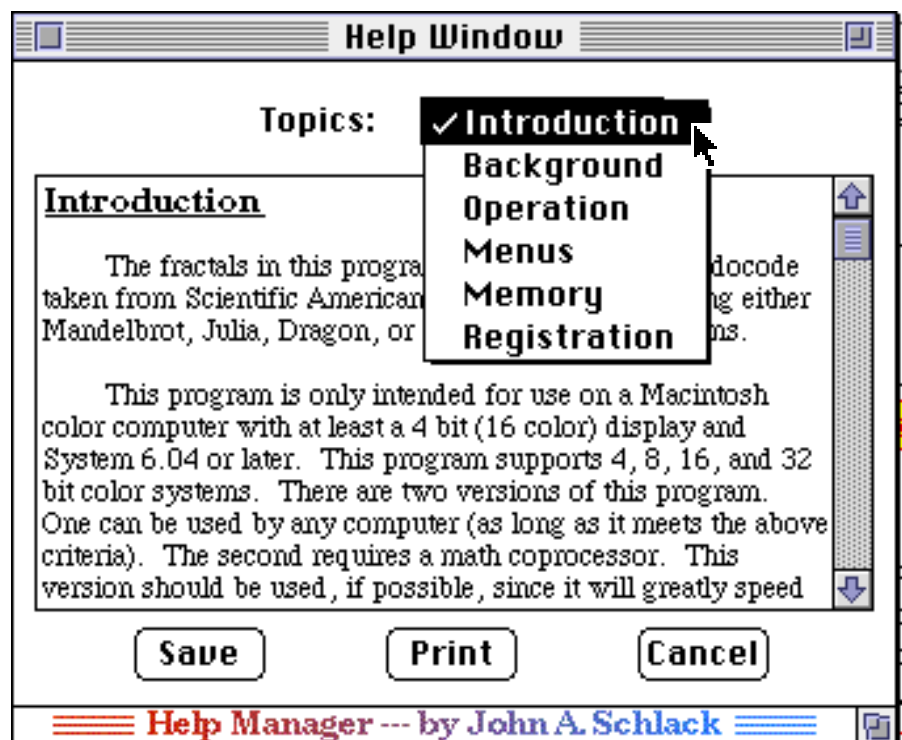


Figure 17: CFG Help Manager Window

A window showing an example of the CFG Help Manager is displayed in Figure 17. There are three basic sections of the window: the topics pop-up menu, the text region, and the buttons.

Information on various topics is displayed by choosing the topic from a pop-up menu at the top of the help window. Selecting a new topic displays its information in the text pane of the *Help* window.

The text pane is the region of the window with the topic information. It can be identified by the scroll bar to its right. Text in this region can be selected and copied, but it cannot be deleted or modified. To print or save the text, use the appropriate button.

The save button will only save the ASCII data (formatting is lost). If a selection is made, only the selection will be saved. If there is no selection, the entire contents of the pane are saved. After pressing the Save button, the standard save dialog box is displayed permitting the user to select the location and name of the data file.

Many applications can open ASCII files. The user can modify the default program that opens the file. To do so, all one must do is open the *Preferences* dialog box and strike the *Creator* icon. A dialog box is displayed which permits either selecting the default application through the Standard Get File dialog box or by manually entering the signature. See the *Preferences* section for more details.

To print the information in the text pane, press the Print button. The help text will be printed with formatting and highlighting intact. Two dialog boxes will be displayed after pressing the Print button. The first is the Page Setup dialog box which permits the user to modify paper parameters. Pressing OK saves the parameters while Cancel will force the program to use default printing options. The second dialog is the print dialog which allows the user to select a page range, number of copies, etc. Selecting the OK button will print the document, while Cancel will abort the printing process. After printing is initiated, it can be stopped by pressing the Cancel button in the progress dialog box (similar to the one displayed in Figure 13). NOTE that the print button always prints the complete information, a selection will have no effect on printing.

One may remove the Help display from the screen by choosing the Cancel button. The help display may also be removed by using the small close box at the left of the window title.

Note that the CFG Help Manager is not a dialog box. Its window may be left up while attending to other activities. If you move the help window behind some other

windows, a quick way to move it to the front is to either select the *Help Window* item from the *Windows* menu or select the *Help* command from the *Apple* menu. The second method will not create a new *Help* window, but simply moves the open *Help* window to the front.

## **Main Menus**

### **Apple**



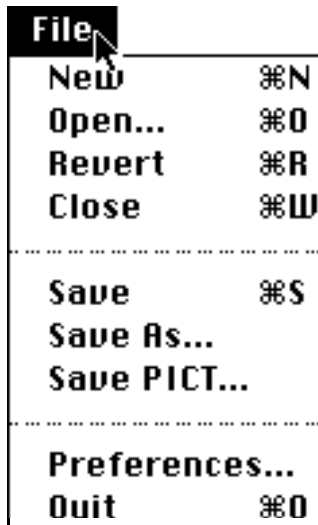
### **About CFG**

The *About* menu item will display a dialog box with information about this application and its creator.

### **Help**

The *Help* menu item will display a window with an interactive help system to provide details about the operation of this program. For a detailed description of the CFG Help Manager, see the Help System section of this document.



**File**

This menu provides general controls for creating windows, saving data, changing preferences, and quitting the program.

**New**

This menu item creates a new fractal with the characteristics described by the *New Parameters* window.

**Open**

*Open* permits the user to retrieve saved fractal files.

**Revert**

Choosing *Revert* restores the current fractal to its last saved state. Possible reasons for disabling this item are: the front window is not a fractal, the current fractal has not ever been saved, or the fractal is being generated.

**Close**

This option closes the front window. If this window is a fractal window, and the data has not been saved, a dialog box will appear asking whether to save the data. Close will be disabled when the front

window is the *New Parameters* window.

**Save**

*Save* puts the data for generating the fractal into a file. It may also store the palette and fractal graphic if these options are set. The program will not save the graphic if it is not a registered version. *Save* is also disabled if the front window is not a fractal window.

**Save As**

This menu item functions just as *Save* does, except it always prompts the user for a file name.

**Save PICT**

*Save PICT* writes the graphic data to a file using the PICT format. Any application that can open color PICT files should open the fractal image. The program will not save the fractal image if it is not a registered version. Possible reasons for being disabled are either the front window is not a fractal window or this program has not been registered.

**Preferences**

This item permits the user to modify execution parameters of this program.

**Quit**

Quits this program. If any fractal open has not been saved, the user will be prompted to save the data.

**Edit**

By using this menu, the user undo a change in the fractal's parameters, transfer data between applications, and modify the palette.

**Undo**

Choose this method to undo the last fractal parameter change made to the front window. If the front most window is not a fractal, the change is made to the global parameters. This menu item will be disabled if no change has ever been made to the fractal parameters. If a change is undone, the Undo command transforms into a Redo command to undo the undo. Subsequent undoes will toggle the menu item between Undo and Redo until another parameter is changed.

**Cut**

This function deletes the selected area from the fractal image and stores it in the Clipboard. Cut will be disabled if no area is selected, the front window is not a fractal, or this program has not been registered.

**Copy**

This function copies the selected area and stores it in the Clipboard. Copy will be disabled if no area is selected, the front window is not a fractal, or this program has not been registered.

**Paste**

This item copies any PICT contents on the clipboard into the selected window region, scaling the image if necessary in order to occupy the entire selected region. Paste will be disabled if no area is selected or the front window is not a fractal. Note that this is not the standard method that paste should function (being disabled if no selection is made and scaling the PICT to the size of the selection).

**Clear**

This routine deletes the selected area from the fractal image. Clear will be disabled if no area is selected or the front window is not a fractal.

**Select All**

This function selects the entire fractal, even if it extends beyond the current window size. This is disabled if the front window is not a fractal window.

**Load Palette**

This function reads in a palette file and applies it to the front window if it is a fractal. If the front window is not a fractal, the palette will become the default palette, being applied to new fractals.

**Edit Palette**

Selecting this item displays a dialog box which permits the user to edit the current palette.

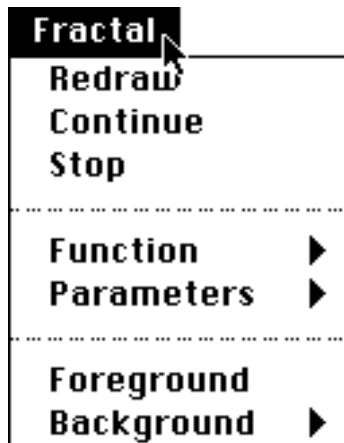
**Save Palette**

Save Palette saves the currently active palette to a file for later use. It can be retrieved by using the Load Palette command.

## Standard Palettes

This is a hierarchical menu for selecting the palette of colors to apply to a fractal. To view the palette, select it and choose the Edit Palette command.

## Fractal



This menu permits the user to choose the fractals to create, edit fractal parameters, and select the fractal generation speeds.

## Redraw

Redraw will start the image creation process from the very beginning. This item may be disabled since the front window is not a fractal or the fractal is being generated.

## Continue

This menu item is similar to Redraw, except that it starts drawing the fractal image from where it left off (if the image generation process had been previously terminated). This item may be disabled since the front window is not a fractal, the previous fractal generation was not canceled, or the fractal is being generated.

**Stop**

This menu item halts the creation of a fractal in the front-most window. It is disabled either because the front window is not a fractal or the fractal is not being generated.

**Function**

This hierarchical menu selects the type of fractal to create.

**Parameters**

This hierarchical menu permits the user to modify the current fractal's parameters or those of new fractals.

**Foreground**

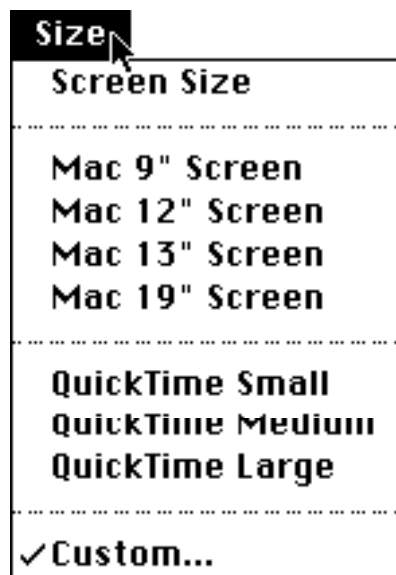
Selecting this item forces fractal creation in the foreground (the user may not perform any other tasks during image creation). This method of fractal creation is the fastest.

**Background**

This hierarchical menu puts the fractal in background generation mode and permits the user to select the speed at which the fractal will be generated.

**Size**





This menu selects the size of the fractal to generate. As the image size increases, so do the memory and time requirements for image generation.

### **Screen Size**

This sets the fractal size to the total space used across all monitors.

### **Window Size**

The fractal image will always fill its window bounds. Be careful, if memory becomes fragmented and the fractal size is increased, the program may run out of memory.

### **Mac 9" Screen**

The image will fill a Mac Classic sized screen (512 x 342 pixels).

### **Mac 13" Screen**

This image fills the standard 13" screen (640 x 480 pixels).

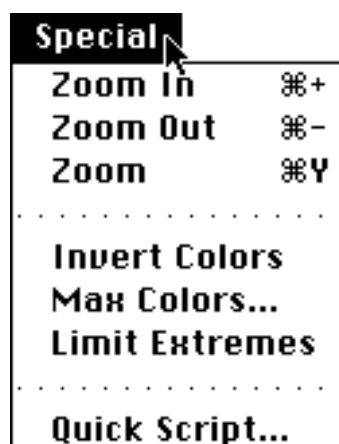
### **Mac 19" Screen**

This image fills the standard 19" screen (1024 x 768 pixels).

### **Custom Size**

This user selects the image size in either pixels, centimeters, or inches.

### **Special**



This menu permits zooming fractals, inverting color, setting the maximum number of colors, and confining the size of random parameters to a smaller area.

### **Zoom In**

This fractal zooms in on the fractal (halving the scale parameters). Image creation is started immediately. Possible reasons for being disabled are either the front window is not a fractal, the fractal is being created, or it cannot be zoomed.

### **Zoom Out**

This fractal zooms out of the fractal (doubling the scale parameters). Image creation is started immediately. Possible reasons for being disabled are either the front window is not a fractal, the fractal is being created, or it cannot be zoomed.

### **Zoom**

Zoom changes the view of the fractal. If a region is selected, the selection becomes the new image bounds. Otherwise, a dialog box is displayed with fractal zooming parameters. Possible reasons for being disabled are either the front window is not a fractal, the fractal is being created, or it cannot be zoomed.

### **Invert Colors**

The colors in the selected region are inverted. The palette will not be affected. Possible reasons for being disabled are either the front window is not a fractal, the fractal is being created, or there is no selected area.

### **Max Colors**

The user may select a maximum number of colors to display. The

palettes will be recalculated with this number in mind. Max Colors will be disabled

if the front window is not a fractal or the program is using true color (32768 or more colors).

### **Limit Extremes**

This function confines the size of random parameters to a smaller area. It is only used if any of the parameters selected are random.

Limit Extreme will be disabled if the front window is not a fractal.

### **Quick Script**

This function permits the user to execute a script in a fashion similar to the UserLand Frontier "Quick Script" window. The menu item is only active when Frontier is running. Note that since the script is executing from a modal dialog box, one will not be able to execute commands targeted for Color Fractal Generator as it is currently busy (running the Quick Script window).

### **Standard Palettes**



Selects the palette of colors to apply to a fractal. To view the palette, select it and choose the Edit Palette command.

### **Grayscale Palette**

This palette is a set of colors consisting only of grays.

### **Rainbow Palette**

This palette has a rainbow of colors.

### System Palette

The standard Mac colors.

### No Palette Loaded / Custom Palette

*No Palette Loaded* means that there is no loaded custom palette.

Otherwise, the name of the custom palette for the active window will be displayed here.

### Function

Function
Mandelbrot
Julia
Dragon
.....
Cosine
Sine
Exponential
Hyperbolic Cosine
Hyperbolic Sine
.....
Tchebychev T
Tchebychev C
Legendre
Laguerre
Hermite
.....
Random Walk

This menu selects the type of fractal to create. The items in this menu will be inactive if the front-most window is a fractal window.

### Mandelbrot

New fractals will be based on the Mandelbrot algorithm.

**Julia**

New fractals will be based on the Julia algorithm.

**Dragon**

New fractals will be based on the Dragon algorithm.

**Cosine, Sine, Exponential, Hyperbolic Cosine, Hyperbolic Sine**

New fractals will be based on one of the complex transcendental functions.

**Tchebychev T, Tchebychev C, Legendre, Laguerre, Hermite**

New fractals will be based on one of the complex polynomial functions.

**Random Walk**

New fractals will be based on the Random Walk algorithm.

**Parameters**

Parameters
Mandelbrot...
Julia...
Dragon...
.....
Cosine...
Sine...
Exponential...
Hyperbolic Cosine...
Hyperbolic Sine...
.....
Tchebychev T...
Tchebychev C...
Legendre...
Laguerre...
Hermite...
.....
Random Walk...

Permits the user to modify the current fractal's parameters or those of new fractals. The menu items will be disabled if the front-window is a fractal window different from the corresponding menu item.

### **Mandelbrot, Julia, Dragon**

Modifies the Mandelbrot, Julia, or Dragon parameters. See Figure 3 above.

### **Cosine, Sine, Exponential, Hyperbolic Cosine, Hyperbolic Sine**

Modifies the complex transcendental functions parameters. See Figure 4 above.

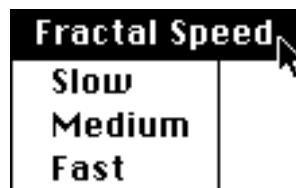
### **Tchebychev T, Tchebychev C, Legendre, Laguerre, Hermite**

Modifies the complex polynomial functions parameters. See Figure 5 above.

### **Random Walk**

Modifies the Random Walk fractal parameters. See Figure 2 above.

### **Background Speed**



This menu puts the fractal in background generation mode and permits the user to select the speed at which the fractal will be generated.

#### **Slow**

The program will generate the fractal in the background slowly. The user will have the most processing power available for other tasks.

#### **Medium**



This speed is a compromise between slow and fast. The fractal is generated faster than Slow, but less processing power is used than in Fast mode.

## **Fast**

The fractal will be generated as fast as possible, but still permitting the user to engage in other activities. It uses much more processing power than Slow.

## **Windows**



This menu permits the user to select a specific window without maneuvering through other windows to find it.

## **Cycle Windows**

This item moves the current window to the back of all windows and makes the next window the current window.

## **Scripts**



This menu is only displayed if UserLand Frontier is running. It enables the user to execute Shared Menu Scripts and edit the Scripts Menu. Three sample scripts are included to demonstrate the power of scripting and how the feature

set of CFG may be easily extended. Detailed information about the sample scripts is located in the scripts

themselves (source code is included - located in the Frontier database) and in the document *CFG Scripting Guide*.

### **About the Scripts Menu...**

This function displays a short message about the Scripts menu. In essence, it says, "This menu contains scripts written with UserLand Frontier."

### **Reset Window Size**

This menu item activates a sample script that resets the size of the top window to the size that it had when the window was first displayed. If the image is smaller than the screen size, the window will enclose the entire fractal. For larger images, the window size will be slightly smaller than the screen size.

### **Stack Windows**

The sample script activated by this menu item stacks the windows such that the bottom window is in the top left corner and each window above it is offset both horizontally and vertically. The *New Parameters* window is made the front-most window. Stacking permits easy access to all open windows.

### **Distribute Prefs**

This sample script copies the preferences data from the *New Parameters* window (actually that global preferences that will be applied to all subsequent fractals) to all other fractal windows. Preferences are not copied to fractals that are currently generating.

### **Edit Shared Menu**

This function makes UserLand Frontier the front-most application and displays CFG's shared menu for editing. One may examine the

sample scripts, add new menu options, or delete old ones.

## **Edit Scripts Table**

This menu item makes UserLand Frontier the front-most application and displays CFG's database table for editing. One may examine (or modify) the low level workings of the glue scripts that transmit the Apple Events.

## **Pop-Up Menus**

### **Open Palette**



This pop-up menu is available in the Get File dialog box that is displayed when *Load Palettes* is selected from the *Edit* menu. This menu permits the user to choose which type of palette to load.

### **CFG**

Selecting this menu item will display only CFG created palettes in the Get File dialog box.

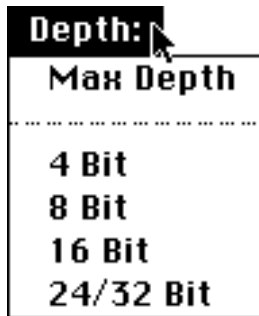
### **'clut' Resource**

This item displays all files. It is used to permit the user to load palettes that were stored as 'clut' resources.

### **'pltt' Resource**

This menu item displays all files. It is used to permit the user to load palettes that were stored as pltt resources.

## **Depth**



This pop-up menu is displayed in the Preferences dialog box. It permits the user to select the color depth of fractals. One may choose a particular depth or generate fractals at the depth of the most colorful screen.

### **Max Depth**

Selecting this option will create fractals that can have as many colors as the most brilliant screen currently attached to the system.

### **4 Bit**

This depth can create fractals with up to 10 colors (4 bit has 16 colors, but 6 colors are defined for other purposes in CFG).

### **8 Bit**

Eight bit depth can create fractals with up to 250 colors (8 bit has 256 colors, but 6 colors are defined for other purposes in CFG).

### **16 Bit**

This depth, usually used with QuickTime, can create fractals with thousands of colors.

### **24/32 Bit**

This is the most demanding depth, requiring 4 bytes to represent for every pixel. It can potentially display millions of colors. Note that although it requires the most memory, generating fractals at this depth

is not much



slower than 8 bit depth. Also, you probably will not notice any difference in image quality if 16 bits is used instead of 24/32 bit depth.

## Scripting

One of the most useful and powerful features in CFG, and something that sets it apart from other fractal programs, is its scriptability. Extensive support has been added to CFG to support UserLand Frontier and AppleScript. CFG also supports shared menus with Frontier.

Scripting is accomplished through the use of Apple Events. A script is written and executed from a client application (such as Frontier or Script Editor). The client application executes the commands in the script, converting any CFG specific commands into Apple Events which are sent to CFG for processing. One may execute scripts from the client application's environment, shared menus (see the *Shared Menus* section below) or the Quick Script dialog box in CFG (*Quick Script* command from the *Special* menu). These last two methods are available only from Frontier.

The Quick Script dialog box (see Figure 18) is similar to the one that exists in Frontier. The user may enter a script of up to 255 characters in this dialog and have Frontier execute the commands by striking the *Run* button. The result of script execution is shown below the words "Response:". The *Done* button discards the dialog.

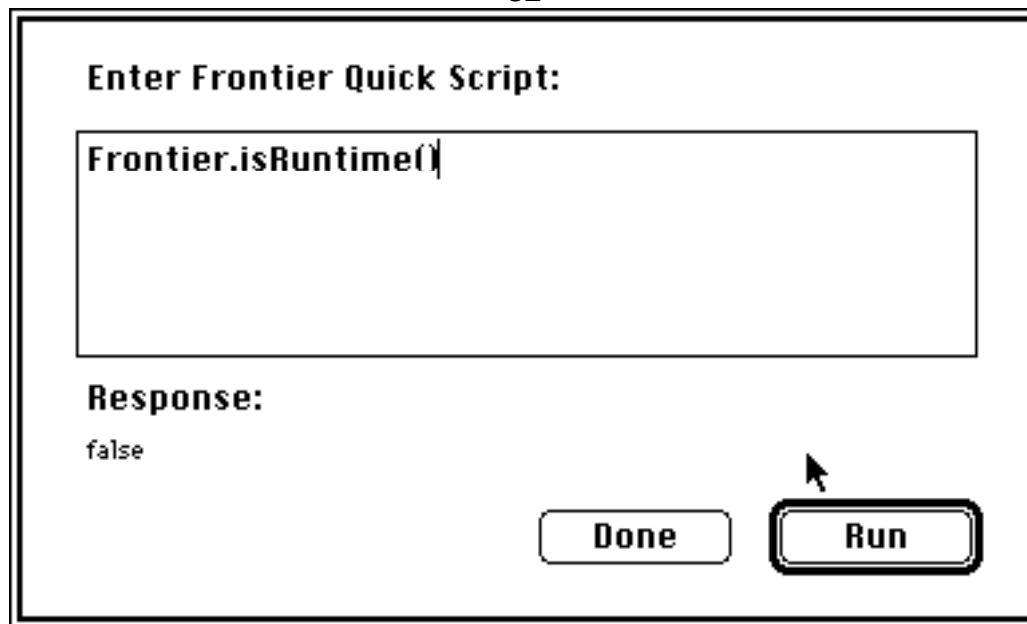


Figure 18: Quick Script Dialog

NOTE that commands targeted for CFG cannot be executed from this dialog box. When the Quick Script dialog box is visible, all Apple Event processing is put on hold (except the events that CFG sends to Frontier to execute the script). Therefore, unpredictable results will occur when trying to execute CFG commands from this dialog box. (Executing CFG commands from the shared menus is perfectly safe).

The data that Frontier needs to support CFG is located in the *CFG & Frontier* folder. All command documentation will be installed in the DocServer database and additional information is located in the *CFG Scripting Guide*.

## Shared Menus

A shared menu is a menu that is appended to the end of the menu bar by another application. CFG supports shared menus from UserLand Frontier. A sample shared menu is already provided - named the *Scripts* menu. The *Main Menus* section above

contains a detailed explanation of the initial menu commands. This menu only appears when Frontier is executing.

The main purpose of a shared menu is to extend the power of CFG. A Frontier script can be attached to the shared menu, permitting the user to execute his favorite scripts without needing to switch to the UserLand Frontier application. Consult the Frontier documentation for information on creating and modifying shared menus. Note that the *Quick Script* command from the *Special* menu also permits one to execute Frontier scripts from within CFG (see the *Scripting* section above).

### Speed Comparisons

Below are two tables designed to provide the user with some idea of the differences in speeds between various program settings and between a standard Mac and a PowerMac. Table 1 shows a comparison of the color depth. The tests compare a Mac IIfx and PowerMac 6100/60 running CFG 2.21 and a PowerMac 6100/80 running CFG 2.3. These tests were executed under the following conditions: Mandelbrot fractal using default settings, background medium fractal generation, Rainbow palette, and an image size of the Mac 9" screen. Note that the direct color depths differ for these tests. This is because my PowerMac only support up to 16 bit color while my Mac IIfx does not support 16 bit color. As can be seen from the table, the PowerMac in 8 bit mode is a speed demon.

Color Depth	Mac IIfx (v 2.21)	PPC 6100/60 (v 2.21)	PPC 6100/80 (v 2.3)
8-bit color	182	10	7
direct color (IIfx - 24 bit,	212	31 (fast direct color)	20 (fast direct color)

PowerMac - 16 bit)                      88 (standard color)    55 (standard color)

Table 1: FPU/Color Depth Speed Comparison (time in seconds)

Notice that the 80 MHz machine running 2.3 is proportionally faster (based on clock speed) than the 60 MHz machine running 2.21. This can be attributed to several factors: gradual addition of extensions over the past 15 months (the Mac IIfx and PPC 60 MHz speed tests were conducted with most extensions off!), the 80 MHz PPC was running System 7.5 while the others ran System 7.1, and a different timing model that makes CFG 2.3 more responsive to the user at the expense of some speed. As the tests became more processor intensive, the speed of the 80 MHz PowerMac started to show.

Table 2 compares the speeds of the three background generation modes (foreground was not considered since this will be phased out in future releases of CFG). These tests were executed under the following conditions: Mac IIfx and PowerMac 6100/60 with 2 MB allocated to CFG, Mandelbrot fractal using default settings, 8-bit color depth, using the math coprocessor, and an image size of the Mac 9" screen. Note that the background times are with the host computer engaged in no other activities. Speeds will slow even more as processor time must be allocated to foreground tasks.

<b>Speed</b>	<b>Mac IIfx</b>	<b>PowerMac 6100</b>
	<b>(v 2.21)</b>	<b>(v 2.21)</b>
slow background	279	15
medium background	182	10
fast background	159	7

Table 2: Generation Speed Comparison (time in seconds)

## **Machines Tested and Bugs**

Version 2.3 has been developed exclusively on a PowerMac 6100/80 with a 256K cache, 24 MB of RAM, and running System 7.5. Version 2.2x has been developed and

tested on a Mac IIfx with 20 MB of RAM running System 7.1 Pro and a PowerMac 6100/60 with 24 MB of RAM running System 7.1.2. All machines were using 32-bit addressing and virtual memory was off. Color depths tested include 4, 8, 16, and 32 bit. I have tested printing only on a laser printer. Script testing was performed using UserLand Frontier 3.02. I cannot guarantee compatibility with earlier versions of Frontier.

If you experience any compatibility problems, please let me know. Include as much system and hardware information as possible, along with a complete description of the bug and when it appears. I'll see what I can do to fix it. I cannot guarantee compatibility for all system configurations (there are just too many, and I have access to very few).

One issue that may arise is if CFG attempts to use a math coprocessor if one is not present. This will lead to a serious and horrendous crash. The only reason for such a case happening is the operating system is informing CFG that a FPU (floating point unit) is present or there is a bug in my code (hey, no software developer is perfect). This crash would most likely occur on startup. You can force CFG to ignore the FPU if you are not afraid of a little ResEdit.

Open a backup copy of CFG in ResEdit. Create a 'Data' resource with an ID of 130 using *Create New Resource* from the *Resource* menu. Type the following hexadecimal sequence: "3638 3030 3000". In the ASCII section to the right of the hex string you should see "680001". Save your changes, quit ResEdit, and launch the modified version of CFG. If all goes well, you'll be greeted by the *New Parameters* window.

### **PowerMacs**

As of version 2.3, the registered version of CFG is a "fat binary", containing

both 680x0 and PowerPC code. The non-registered version contains only 680x0 code, which

will execute very slowing on a PowerMac. To compensate, the PowerMac version of CFG 2.21 (*powerCFG*) is included to demonstrate the speed.

Versions with PowerPC native code are FAST!!! They consistently outperform a Mac IIx, usually by 1500% to 2000% in 8 bit mode. For example, when generating a 100 frame PICS sequence (160 x 120 pixels, 8 bit color) using the "Movie Demo" Frontier Apple Event example, the Mac IIx required 2 hours and 12 minutes. A PowerMac 6100/60 (no cache) finished in under 8 minutes. That's at least 16.5 times faster!

A word of caution is advised, however. The huge speed increase shown above was not nearly as great when the 6100 was run in 16-bit color mode. The speed degradation is due the time required for the operating system to perform color conversions and the additional video data that is moved with 16-bit color. Every attempt will be made to reduce this bottleneck. The speed degradation can be seen in Table 1 (6100/60 is 18 times faster than a IIx in 8 bit mode but only 7 times faster in direct color mode). One can minimize the speed decrease for the Rainbow palette in direct color mode by selecting the *Use Fast Direct Color* option (see the *Preferences* section).

**User of "original" PowerMacs:** the transcendental math calculations routines in the original ROMs (in 6100s, 7100s, and 8100s) were quite slow. Apple released an enhancement shared library called *MathLib* (I believe the version is 2.0). This speeds transcendental calculations considerably. I cannot release it with the code, but be sure to locate a copy and place it in your Extensions Folder.

### **Registration**

Please don't skip this section. Even if you don't plan on registering initially,



some of what I have to say might entice you to register, especially if you are interested in obtaining the source code to this program.

Quick Price List

standard registration:	\$20.00
standard upgrade from CFG 2.2:	10.00
source code registration:	45.00
outside the continental US add:	3.00

Standard Registration

This program is Shareware. I am asking for a registration fee of \$20 for the Standard Version of CFG. If you pay this fee, you will receive the registered version of the program for all Macs and a version accelerated for PowerMacs. These programs will not prompt for registration. You will also receive the skeleton **code for creating your own CFG palettes**. All you need to do is put in the code to create the desired colors. The source is in THINK C 6.0 format.

Source Code

Ok, now, here's the real value. If you send \$45 for registration, I will send you the Master Version of CFG. **Learn to program the Power Macs!** This includes the **FULL SOURCE CODE** in Metrowerks CodeWarrior C format. Only slight modifications are required for it to compile on THINK C 6.0 or later. The code includes portions of Frontier SDK 3.0 which are used to build CFG (source has been modified to work with the Power Mac). I highly suggest that you obtain the entire SDK as it will prove to be a valuable addition to your development library. This will allow you to modify the procedures for your own program (however, you are not permitted to release or sell modified versions of this program or release or sell code which is based entirely or in part on this program's source code). You also will receive all the benefits of the Standard Version described above plus a free upgrade when the next version becomes available

(source code not included for the free upgrade).

### Upgrades

Registered users of version 2.2x of CFG may receive an upgrade to the current version for \$10. If you are a registered owner of the source code, contact me about source code upgrades.

### Outside Continental U.S.

Please add \$3 for shipping and handling.

### General

Please, when you register, try to send either a check or money order. I will accept cash, but it could become a real hassle, especially if it gets lost in the mail. Mention that you are registering for Color Fractal Generator 2.3 (I have a couple other Shareware products). Finally, if you have an E-mail address, provide that as well.

Send your Shareware fees to:

**John Schlack**  
**1411 Hi View Drive**  
**Southampton, Pa. 18966**  
**USA**

### **Feedback**

Register and give me feedback. I want to know what you think. What do you like? What do you want improved? Added? Deleted? Overhauled? Registration is \$20. You will receive both the 680x0 and PowerMac versions of CFG. For \$45, you get all the above, one future update free, plus the CodeWarrior source code (for the first version only). A great value considering the time that I

put into this program. Send your comments and registration fees to:

**John Schlack**  
**1411 Hi View Drive**  
**Southampton, Pa. 18966**  
**USA**

Check out my web site (World Wide Web):

<http://www.voicenet.com/~jschlack>

You can also reach me through E-mail:

Internet: [jschlack@voicenet.com](mailto:jschlack@voicenet.com)