
A terrain manipulation thingy



Version 1.39

jslayton@ridgenet.net
http://www.ridgenet.net/~jslayton/software.html

# 1. General

## 1.1. Introduction

Why "Wilbur"? Very simple: I write software, I don't do marketing. It was the very first name that came to mind after the decision to do a public release.

Wilbur is a simple height field toy. It loads, processes, and saves height fields. It can calculate texture maps based on the height field data and color the map so that it shows both height and slope information. It can also be used to create synthetic height fields.

It stores the height field points as single-precision floating-point values and texture information as 24-bit RGB values. What this means is that the surface typically requires about 9 bytes per sample. For a 1024x1024 image, that's about 9 MB of RAM required (undo requires a like amount of memory for each undo level). The software will take advantage of virtual memory, but expect overflows to disk to slow things down significantly.

Wilbur started life on a 33 MHz 80486 processor with 16 MB of RAM. It has some features that date back to that early period, but most of the limitations dating back the full 10 years have been removed in recent versions.

## 1.2. Legal Bits

Due to the actions of certain individuals, we now a need to clear up any confusion regarding the rights and responsibilities of both those who use Wilbur and me, who wrote Wilbur.

The executable and all documentation are copyright 1997-2005, Joseph Slayton. This software (and documentation) may be freely copied and distributed provided all copyright notices remain intact. It may not be sold, but a nominal fee may be charged for media and handling by distributors if required. It may not be given away as part of a commercial package without the explicit written permission of Joseph Slayton. I make no warranty warranty of this software's suitability for any purpose, and I accept no liability for use or misuse of the software.

The quick summary of all of the above: You can't sell this software because it's not yours. You can give it away, but if you want to give it away to increase the value of something you're selling, you need to get my permission first. I do not say that this software can do anything at all, even run. If you run this software and bad things happen *it's not my fault*.

If you do not agree to these conditions, you may *not* use the software and must eliminate all copies in your possession.

## 1.3. System Requirements.

The following is the minimum system configuration needed to run Wilbur:

- 450 MHz Pentium-class CPU
- 64 MB RAM
- 4 MB free hard disk space
- 16-bit or 32-bit color display adapter
- Windows XP (95/98/Me used to work, as did NT4 and 2000).

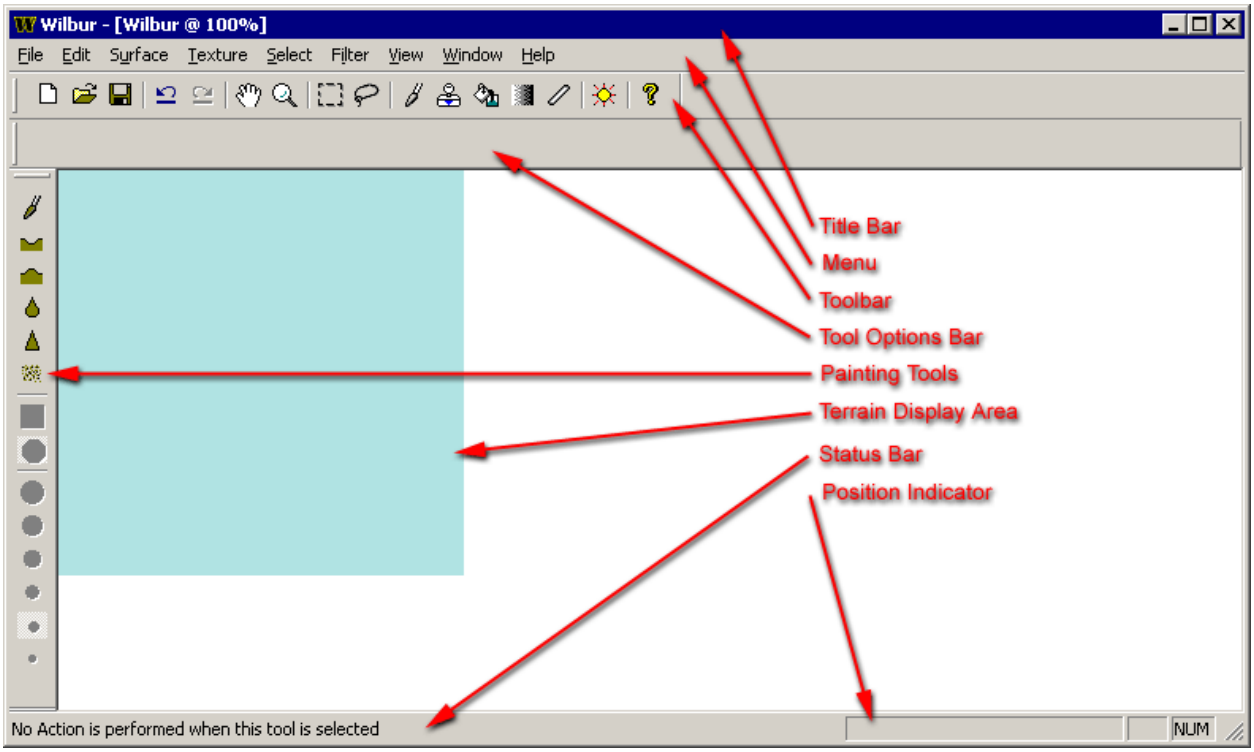
As with all things in the PC world, more is better. Wilbur doesn’t take advantage of the fancy graphics hardware of modern machines (yet), but can always benefit from more memory. Note that Wilbur probably still works on 98, Me, NT4, and 2000, but I can’t say for sure. 95 is right out.

## 1.4. What's New

See the newest section of Credits and history: for full details.

# 2. Main Window

The figure below shows an example of the main Wilbur Window. This is a picture of how Wilbur would appear in initial startup before any tools are selected and before any operation is performed.




The Title bar shows the program name, the name of the current document, and the displayed zoom level.

The Menu is a simple old-fashioned Windows-style menu bar. The individual menu items are described in section 3.

The Toolbar contains shortcuts to some of the most commonly-used menu items.

The Tool Options Bar shows the operations for the currently-selected tool. In the image above, no tool is selected.

The Painting tools are all shortcuts to options in the Custom Brush. These options are described in section 5.

The Terrain Display area shows the current terrain image.

The Status Bar provides feedback about the current state of the system Such feedback includes longer descriptions of menu items as well as information about how to use the currently-selected tool.

# 3. Menus

## *3.1. File*

This menu is roughly the standard Windows menu (open/save/exit). Most file operations are handled via the open/save item. The operations that deal with multiple types of input file (such as SRTM or DTED input) are handled via the import menu. A quick summary of which file types the system supports and how it supports them (texture map, height field, read, and/or write) is given in the table below.

| | | | Height Field | | Texture Map | |
|---|---|---|---|---|---|---|
| Dialog File Type Label | Description | Extension | Read | Write | Read | Write |
| Grayscale Image Surface | Load image and construct height from grayscale composition of the color channels. | BMP, PNG, GIF, JPG, etc. | Yes | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Color Image Texture | Load image into texture channel. | BMP, PNG, GIF, JPG, etc. | | | Yes | |
| PNG | Portable Network Graphics | PNG | Yes | Yes | Yes | Yes |
| MDR Surface | Custom surface file format. | MDR | Yes | Yes | | |
| Muse DTED surface | 16-bit MUSE DTED | DTE | Yes | Yes | | |
| USGS DEM surface | 16-bit USGS DEM text format | DEM | Yes | | | |
| Matlab MAT 1.0 | MATLAB floating-point format. | MAT | Yes | Yes | | |
| Raw DTED cell surface | Level 1 DTED-format one degree cell. | DT1 | Yes | | | |
| Terragen | Terrain file. | TER | Yes | Yes | | |
| VistaPro DEM | | DEM | Yes | | | |
| PCX Image | 8 or 24-bit PCX file | PCX | Yes | Yes | Yes | Yes |
| Windows BMP image | 8 or 24-bit BMP file | BMP | Yes | Yes | Yes | Yes |
| TARGA Image | 24-bit TGA image | TGA | Kinda | POV | | Yes |
| TARGA surface | Fractint-format 16-bit surface data in 24-bit TARGA file | TGA | Yes | Yes | | |
| STM Surface | | STM | Yes | | | |
| Binary Terrain Surface | Vterrain.org's BT file format. Doesn't support the projection info | BT | Yes | Yes | | |
| Ray Dream Studio 4.1 | Ray Dream Studio 4.1 triangles | RD4 | | Yes | | |
| POV Include File | POV-Ray smooth triangle mesh include file. | INC | | Yes | | |
| 3D DXF Quads | DXF quads that represent the surface. | DXF | | Yes | | |
| RAW triangles | RAW triangles that represent the surface. | RAW | | Yes | | |
| 16-bit PGM | PGM 2-byte file | PGM | | Yes | | |
| Lon/Lat Mesh | Simple text file with lon, lat, alt; one point per line | TXT | | Yes | | |
| 2D DXF Contour Map | Data EXchange Format file of lines representing level contours in the map. | DXF | | Yes | | |
| 2D CC2 Contour Map | Campaign Cartographer 2 file of lines representing level contours in the map. | FCW | | Yes | | |
| 2D SVG Contour Map | Scalable Vector Graphics file of lines representing level contours in the map. | SVG | | Yes | | |
| Wavefront OBJ | Wavefront OBJ file made of triangles for the surface. | OBJ | | Yes | | |

### 3.1.1. New

Creates a new surface and asks for information about the size and edges. Getting data into the file requires either creating a new surface (Surface→Calculate Height Field) or reading one from a file (File→Open).

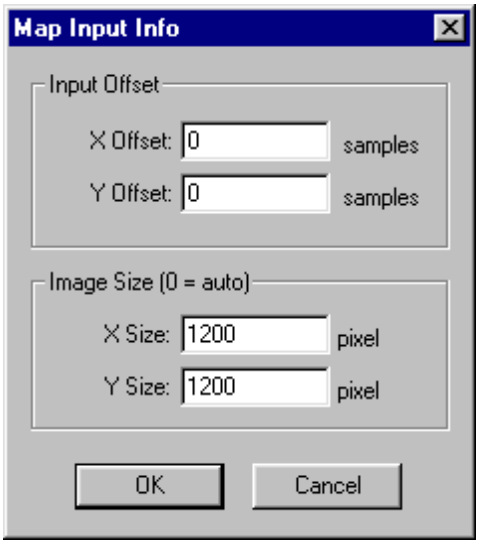
### 3.1.2. Open

Calls the standard Windows Open dialog. The file formats are summarized above and discussed at length below. The type of file is always taken to be the type of file displayed in the dialog. This feature allows a file without the extension (say a .BIL file that needs to be opened as a .BIN file) to be opened properly. Due to mindless, lazy programmers, however, the program isn't really hot in the type detection department and always uses the type shown in the type drop list. *Always* select the appropriate type from the drop list. Really. I mean it. Many of the read operations will fail with no apparent error (just a real quick operation) if the wrong file type is selected. It's even possible to crash the program.

#### 3.1.2.1. Muse DTED 16-bit binary surface

The Muse data format is unusual, to say the least. It does, however, contain a header that allows storage of the terrain boundaries and other pertinent information. This was another customer requirement that survived from some ancient software. The Muse DTED format has nice header information, but the variant supported here has one major difficulty: it only supports integral values. What that means is that if your surface is consists entirely (for example) of values between 0.1 and 0.9, the system will save the file as consisting entirely of 0 and 1 values. This format is good for "normal" terrain, but loses something on almost flat surfaces.

If the "Prompt for trim on DTE Read" check box on the Misc Options dialog is checked, then the dialog shown below appears.




The X Offset and Y Offset fields give the offset *in pixels* from the upper left edge of the file to start reading. The X Size and Y Size fields specify the width and height of the data area to import. If the X Start + the X Size or Y Start + Y Size is greater than the width and/or height of the imported file, bad things can happen. You have been warned.

#### 3.1.2.2. TARGA surface (POV height field format)

The POV raytracer has a 16-bit height field format that takes advantage of TARGA files. Wilbur can import and export surfaces in that format. There are currently no facilities to generate POV source files to render the data.

#### 3.1.2.3. TARGA Image (one of 3 color channels to height)

Any TARGA file may be imported as a texture map for the surface, but only one channel may be used. You will need to pick the red, green, or blue channel as desired.

#### 3.1.2.4. USGS DEM Surface

The US Geologic survey has data sets for the US available on the Internet. These DEM files are large, ugly text files. The software has been tested on 1-degree by 1-degree and some 7.5 min files. The reader isn't terribly robust, but it's worked with all of the files from the USGS I've tried it on so far. Note that the 7.5 min files are UTM-based; because of this basis, they don't have the proper edges after the import.

#### 3.1.2.5. DMA DTED Level 1 single cell

This software can also read a single DTED cell. Not overly useful, but it does come in handy from time to time.

#### 3.1.2.6. BMP Texture

Wilbur supports the ability to read a texture map into the display portions without having to load a height field. This feature is particularly useful when attempting to reproject a map file. For example, if a rectangular-projection map of a planet is available (say, a picture of the earth or Mars), this file could be read in as a texture map and the Misc→Map Projection dialog could be used to view the map in another projection and output at a different resolution, without having to have a height field present. If the file isn't a multiple of four in width and height, the file will be cropped until it is (losing at most three pixels from top and/or bottom).

#### 3.1.2.7. MATLAB MAT File

Supports reading of Intel-byte ordered (only) single or double-precision floating-point. This format is the same type that can be processed by GFORGE and HFLAB. Note that if the input file is a complex-valued matrix, only the real portion will be read in; the complex portion will be ignored.

### 3.1.3. Save As

This software has no Save menu item, only Save As. A name must always be specified. The types of files that are supported are described below.

#### 3.1.3.1. BMP Texture

Saves the current texture image (not the surface data) as a 24-bit BMP file. Not much else to be said, really.

#### 3.1.3.2. PCX Texture

Saves the current texture image (not the surface data) as a 24-bit PCX file. Does anybody use PCX files anymore?

#### 3.1.3.3. TARGA Image

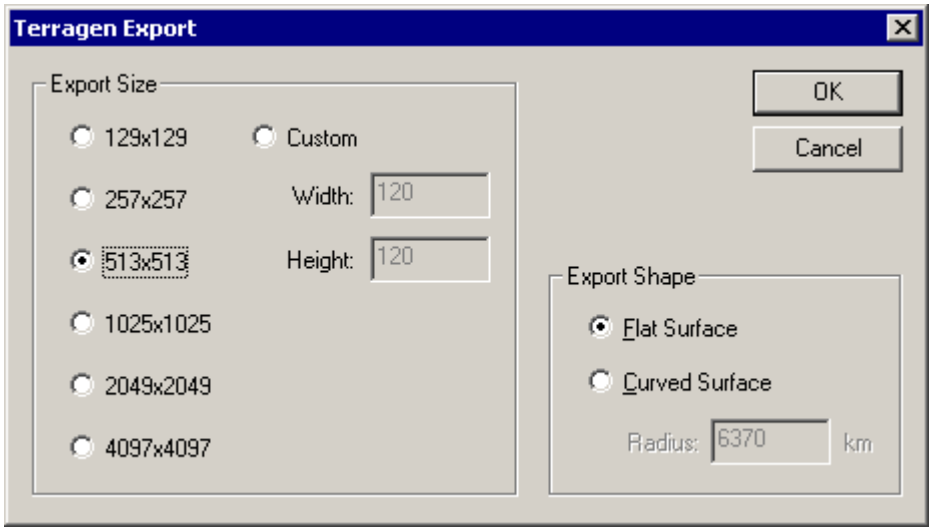
Saves the current texture image (not the surface data) as a TARGA file with a depth equal to the current color depth. For 8-bit color, each pixel will be expanded to the full 24-bit palette entry in the data file; for the 24-bit texture format, it will be the contents of the texture map.

#### 3.1.3.4. TARGA Surface (POV height field format)

See the entry on Open, above for a description of POV TARGA surface files.

#### 3.1.3.5. Terragen Surface

Terragen is a spiffy terrain renderer. Wilbur can output terrain files for use with Terragen. It can also read those files, too! The export dialog looks like this:




Export Size indicates the output size of the terrain. The items in the left column are preset sizes; Custom can be used to set a width and height not found on the list.

Export Shape indicates the shape of the render. Terragen supports both flat and curved height fields. Curved height fields provide items that move below the horizon.

The Wilbur export assumes that the terrain data is set at a 30 meter spacing (the Terragen default). It does not provide a way to change this interval.

### 3.1.3.6. Muse DTED 16-bit binary surface

See the entry on input formats, above, for a description of Muse files.

### 3.1.3.7. 16-bit binary surface

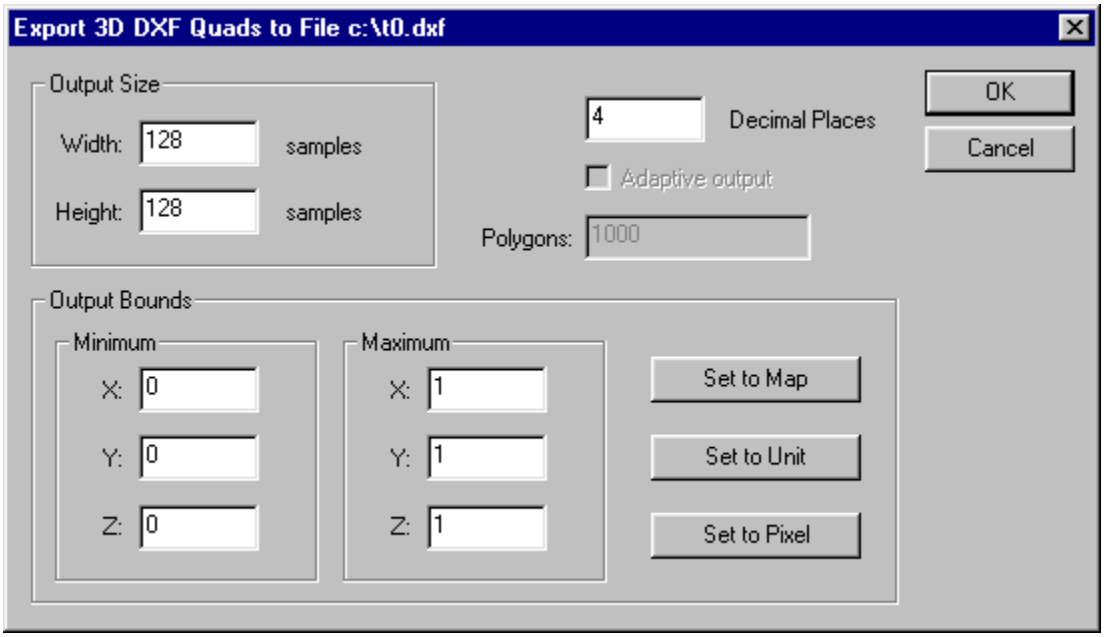
The current support for binary file output is no header, 16-bits per sample (values in surface rounded down to nearest whole unit), local byte ordering. A future upgrade will be the reverse of the binary load operation.

### 3.1.3.8. 3D DXF Quads

All 3D DXF output is done as quads on a uniformly sampled grid. The corners for the output quads are sampled evenly across the surface, interpolating between data points where necessary. No averaging is performed when a large surface is sampled to a smaller.

The DXF output is scaled to fit in a unit cube. That is to say, the edges of the DXF field are 1 unit long and 1 unit wide and the height field is scaled so that it is 1 unit high. This scaling holds true regardless of the sampling resolution or number of samples in each direction.

The dialog below shows the options.




Width and Height are the desired resolution of the output data. Be warned that the DXF output files are very large, taking about 168 bytes for every point output, and that Width*Height points will be output. Thus, a 100x100 surface (10000 data points) gives a file 1,680,102 bytes long.

Decimal Places shows the number of decimal places to go into the coordinates in the output file.

Adaptive output and Polygons will be implemented when adaptive surface processing comes about in a future release.

Output Bounds indicate the scaling that will be used to output to the data file. The default values are a unit cube: the X (left-right) values go from 0 at left to 1 at right, the Y (top-bottom) values go from 0 at bottom to 1 at top, and the Z (height or in-out) values go from 0 at the lowest point to 1 at the highest point. This output can be scaled to any size desired using the minimum and maximum values.

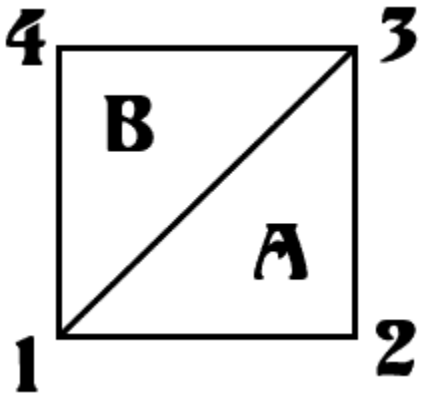
The buttons in the Output Bounds box are provided to simplify setup for the most commonly desired output scaling settings. The functions of the buttons are (clicking button sets indicated values):

| Button | Minimum X | Minimum Y | Minimum Z | Maximum X | Maximum Y | Maximum Z |
|---|---|---|---|---|---|---|
| Set to Map | Map left | Map bottom | Lowest alt | Map Right | Map Top | Highest alt |
| Set to Unit | 0 | 0 | 0 | 1 | 1 | 1 |
| Set to Pixel | 0 | 0 | Lowest alt | Map Width-1 | Map Height-1 | Highest alt |

### 3.1.3.9. 3D Raw triangles

Some software can handle a file consisting of simple raw triangles. Wilbur outputs the data in this more compact format as well as the DXF format. The dialog looks exactly like the 3D DXF one shown above except that the title says "RAW" instead of "3D DXF Quads". All of the items mean the same thing.

Each 4 points on the surface generate two triangles. The picture below shows the numbering of the vertices. Triangle A is output first, with vertex ordering 1, 2, 3. Triangle B goes out to the file next, with vertex ordering 1, 3, 4.




A RAW file consists of a simple text format that contains one triangle definition (three vertices separated by spaces) per line. Each vertex consists of three numbers, separated by spaces. The output isn't exciting, with each triangle getting nine numbers on a line.

### 3.1.3.10. 3D TIN

The triangle irregular network file type is used by some programs, so we support it. It is functionally identical to the RAW file format and very similar in structure. It is a simple text format, with one triangle record per line. Instead of just numbers, each line starts with a 't'. after the 't' are the three vertices, with a space between each number in a vertex and three spaces between each vertex definition. While this definition may not be the "correct" definition for a TIN file, it works well enough for the program I've got that wants TIN input.

### 3.1.3.11. Ray Dream Studio 4.1

MetaCreations makes the Ray Dream Studio 3D modeling program. Native file format is provided for the 4.1 version (which also works for the 5.0 version). The triangles used are defined in the same way described in the 3D RAW Triangles section, with the parameters described in the 3D DXF Quads.

### 3.1.3.12. POV Mesh Include File

The POV-Ray raytracer is an effective product at an inexpensive price (free from http://www.povray.org). Export is a triangle mesh with smooth normals (the four adjacent samples in X and Y area used for normal computations). This file will be included in a POV-Ray definition file for final rendering. Note that a texture is not specified for the terrain, but a commented out texture statement is found at the end of the file. The mesh definition is named "TerrainMesh" by default.

### 3.1.3.13. 16-bit PGM

The scape variant I'm using to reduce the complexity of the surfaces for some terrain flybys needs a simple, 16-bit PGM format file. As might be expected, Wilbur supports this feature. The 16-bit PGM file format used is a simple text file with the first line consisting of something like "P2 width height max" where 'P2' is the header, 'width' is the numeric width of the surface in samples, 'height' is the numeric height of the surface in samples, and 'max' is the highest point in the surface. Following this line is the data, with numeric values separated by spaces, no more than ten to a line. This is a somewhat inefficient form to handle the data, because the file will likely expand when stored to disk. Also, the system clips height values to 0 at the low end (all values less than 0 become 0), max at the high end (or 32767, whichever is less), and quantizes all data to integers (all fractional heights are truncated to the lower integer value).

### 3.1.3.14. MATLAB MAT File

Supports output of Intel-byte ordered (only) single precision floating-point. The matrix is real-valued only. This format is the same type that can be processed by GFORGE and HFLAB.

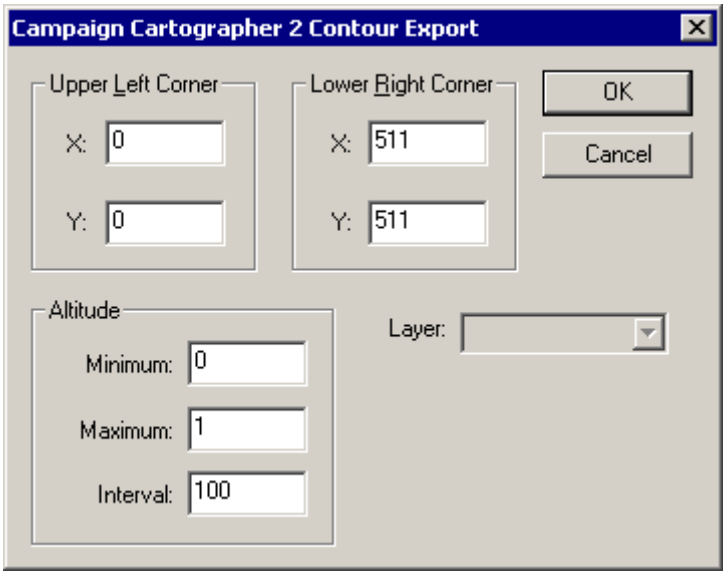
### 3.1.3.15. Lon/Lat Text Mesh

Outputs a simple file consisting of the lon value, lat, value, and alt value of each point specified on the surface. Uses the same dialog box as the 3D DXF Quads, Raw Triangles, etc. to specify the resolution of the surface. One point is stored per line, with spaces between the data items. Order is lon (X), lat (Y), alt (Z) from left to right across each line.

### 3.1.3.16. 2D Contours (DXF, CC2, and SVG)

The 2D contour output types use the same dialog. The DXF and CC2 files output unfilled black unclosed contours, while the SVG file is output as unfilled colored unclosed contours.

Selecting this menu option brings up the dialog shown below:




The Upper Left Corner and Lower Right Corner values represent the scaled output area in the file. The whole surface currently in memory will be saved to the file with the bounds indicated. These controls can be used to adjust the aspect ratio of the data file, if desired. The output file is generated on the standard 1000'x800' template, regardless of the actual map dimensions. Note that the contours are based on a full-resolution image: a high-resolution image can result in a large contour file.

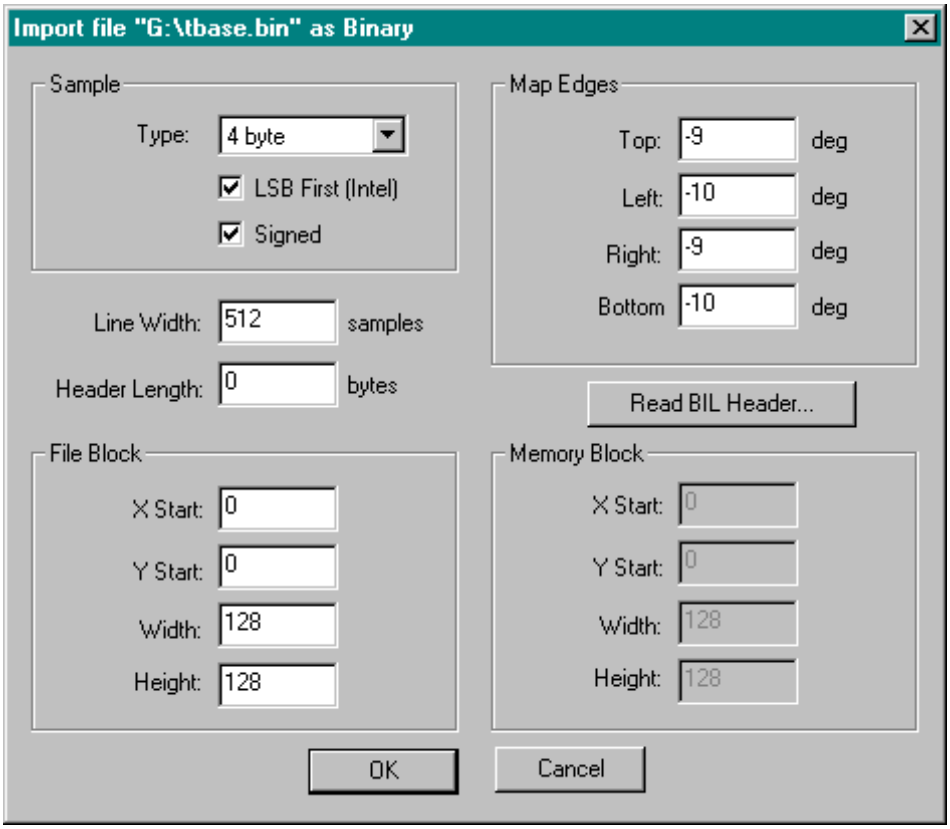
The Minimum, Maximum, and Interval values indicate the vertical extents and resolution for the output. The first contour starts at Minimum and then outputs every Interval units until the level is greater than or equal to Maximum. As an example, a coastline contour map can be generated by setting Minimum to 0, Interval to some value (100 works well), and Maximum to a value less than Interval (50 works well).

## 3.1.4. Import

The Import menu provides support for options which don't fit easily into the current Open scheme of things.

### 3.1.4.1. Binary Surface

Binary surfaces can be imported from almost any uncompressed source using 1, 2, or 4 byte data samples. The dialog shown below obtains the required data for imports.




***Sample*** deals with the individual samples in the file. The program can handle 1 byte, 2 byte, 4 byte, and 4 byte floating-point formats. The integer samples can be signed or unsigned as indicated by the Signed check box and can be interpreted as LSB first (Intel style words) or MSB first (Motorola style words) as indicated by **LSB First**.

**Line Width** is the width across a line, in data samples, There is currently no facility to handle lines that are not an integral number of samples wide. This option is used because the load doesn't always read the full data across a line and some software has lines that aren't filled with data.

**Header Length** is the number of bytes to skip before reading data.

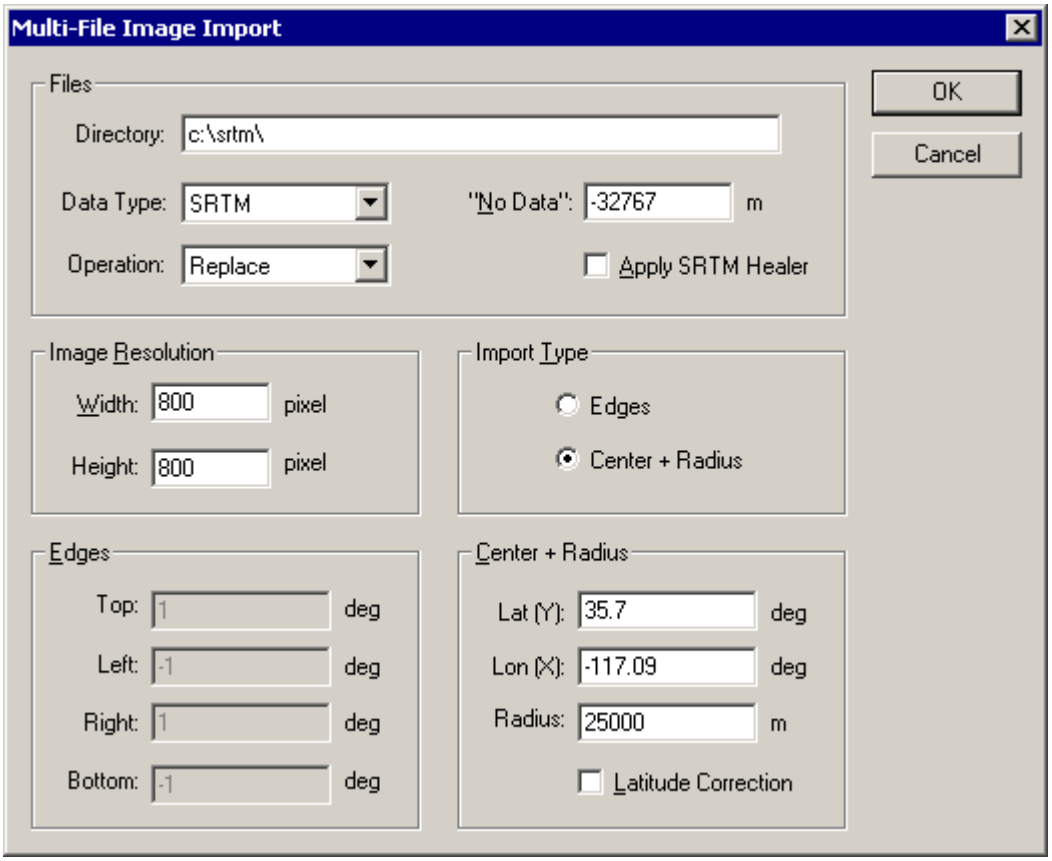
***Map Edges*** gives a quick way to respecify the bounding box for the data area. You must specify this value if you know the intended location. Importing a small area of the surface uses the full bounding box specified.

***File Block*** gives the location and size of the area in the source file to import *in pixels*. Again, **X Start** + **Width** and **Y Start** + **Height** must be less than or equal to the file size or bad things can happen.

***Memory Block*** specifies the destination area of the data from the file into the surface in memory. These parameters are grayed-out in the Open dialog because the surface is automatically resized to fit the data.

### 3.1.4.2. Multi-File Image

Imports a surface from one or more NGA DTED Level 1 CD-ROMs or from multiple data files from the Shuttle Radar Topography Mission (SRTM) . Will handle imports across multiple cells as appropriate. If you have DTED or SRTM data, this one is fun. SRTM data can be obtained via the internet, while DTED data is available from the US military.




**Files** indicates where the source data is located and what process to take with the data. Directory is the directory where the DTED CD-ROM root or SRTM data is stored. Data Type is the type of data stored in the specified directory. Operation indicates how the data will be processed relative to the existing surface in memory. ”No Data” is the value that will be used when the import does not contain required pixels. Apply SRTM healer will use a very simple healing algorithm to reduce artifacts from SRTM imports that are missing pieces (very common).

**Image Resolution** will be active if operation is Replace. It indicates the size of the imported image, in data samples.

**Import Type** indicates how the edges of the map will be specified.

**Edges** is used when Import Type is set to Edges and directly specifies the edges of the map to be imported.

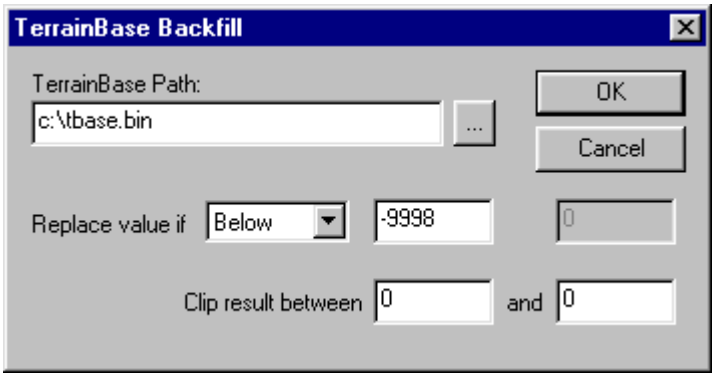
**Center + Radius** is used when Import Type is set to Center + Radius and specified the center and the radius of the imported area. Latitude Correction will distort the map horizontally so that the center of the map is at a 1:1 scaling ratio (the same ratio as the height). These imports are usually distorted by a factor of 1/cos(latitude).

### 3.1.4.3. Image Subsection

The Image subsection command is conceptually quite similar to the Binary Open command. It differs in that instead of resizing the surface data to fit a file read, it reads data from the file into the image already in memory. See the section above on binary imports for further details, except pretend that the Memory Block section isn't grayed-out.

### 3.1.4.4. TerrainBase Backfill

There are times when dealing with a real-world terrain that the normal surface won't have certain sections of information. For example, GTOPO30 data sets don't have information about the sea floor. TerrainBase, on the other hand, has some information about seafloors, although at low resolution. This option allows missing parts of a surface to be filled in with data from the TerrainBase data set. There is a full discussion in the tutorials about using this operation.




**TerrainBase Path** is the path to the TerrainBase data file (or any other 4320x2160 sample, Intel order, 16-bit-per-sample data set) that will be used to fill in the missing parts of the surface.

The **Replace value if** drop list has three options: Below, Above, and Between. The two values to the right of the list will hold the low and/or high point that determine the vertical extent(s) to be replaced.

**Clip result between** takes two parameter, the first of which is the low point, the second of which is the high point. If the result of filling the surface is outside of the range indicated, the value will be clipped to the range. The TerrainBase data set doesn't exactly match many of the higher-resolution data sets (GTOPO30, for example), so this option allows the parts that would be obviously wrong to be removed.

#### 3.1.4.5. World Builder Map

Dave Allen wrote a program for Windows called World Builder some years back. It had very nice tectonic effects and generally nice results. I wrote an import filter to read the World Builder maps and reproject them into an Equirectangular projection for use with the Map Projection window in Wilbur. World Builder seems to have disappeared, but this feature remains.

### 3.1.5. Export

The Export menu primarily exists to support weird processing options that I have used to support my customers at my day job. I include these options in the general Wilbur release because a) my customers can use Wilbur to process their own data, and, b) I'm too lazy to take them out.

#### 3.1.5.1. MONSYS Textures

The MONSYS Textures option, very simply, reads multiple DTE or DEM files and outputs the image maps as BMP files. I have another program that handles large areas of terrain images as background for system data. It uses BMP files for the background images. I wanted them all to be calculated with the same lighting parameters, but I had gotten very tired of processing them one by one. So I set up this menu option. Eventually, it will probably mutate into a batch processing option that will read a sequence of pretty much any files and output the images as a sequence of any type of image file it can handle. Not important now, though.

#### 3.1.5.2. RangeView Terrain MIFs

The RangeView display system has a peculiar data format that it needs to use. Wilbur can output multiple 1-degree cells from SRTM or DTED data to generate files for use with this product. The output isn't the best, but it does work after a fashion. Note that Wilbur can also save the current surface as a terrain MIF.

#### 3.1.5.3. Process GMS Tile File

Reads a GMS map tile file (MONSYS V2) and generates

### 3.1.6. Exit

Quits the program.

## 3.2. Edit

### 3.2.1. Undo

The Undo command undoes the last painting or whole-surface operation. Note that certain kinds operations (such as the texture operations and changing lighting or other settings) cannot be undone.

Be warned that the undo/redo option is a potential memory pig. This problem arises from the fact that to perform an undo, the system saves the *entire* surface, image map, and selection to local memory. The amount of memory required is directly proportional to the number of undo levels desired and the size of the surface.

### 3.2.2. Redo

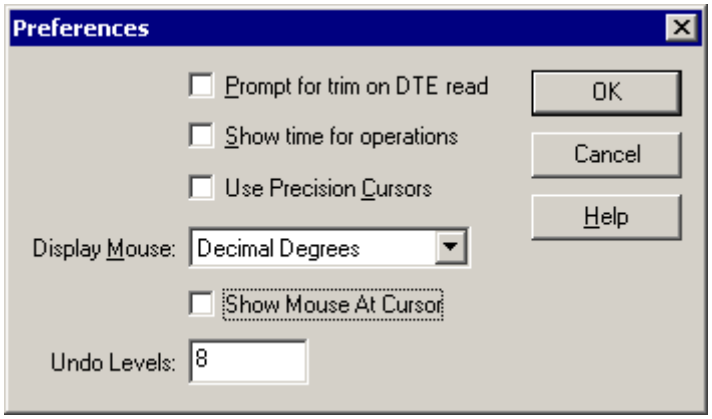
The Redo command undoes the last undo operation. This menu item will only be available if the "Enable Undo" option in the Misc→Options is set and if there has been an undo since the last change.

### 3.2.3. Fade to Prior

Fades the current surface with the top item on the undo stack. This operation allows the strength of an operation to be controlled, if in a somewhat clumsy way.

### 3.2.4. Preferences

Preferences contains some of the miscellaneous settings that didn't have a home elsewhere. This dialog is shown below.




Prompt for Trim on DTE Read only has an effect when reading a Muse DTED format file. It allows a specific part of the file to be extracted.

Show Time for Operations will show the time between when the system puts up a progress dialog to show the progress of an operation and the time it takes down the dialog. This option can be useful for timing simple operations, but gets highly annoying if you're using an operation that's composed of multiple passes, because each pass will be timed and you'll have to click OK for each.

Use Precision Cursors will cause the painting, cloning, and related cursors to take the form of a '+' instead of their normal pictures. These cursors make it easier to precisely place the cursor in a desired location onscreen.

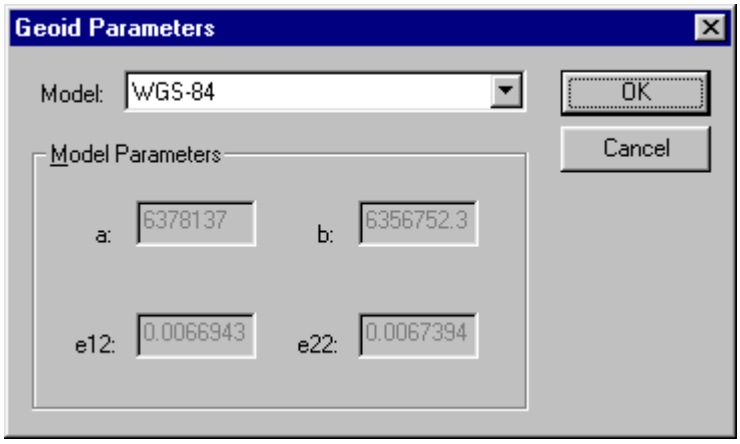
Display Mouse allows the type of display representing the mouse position that will be shown in the status bar to be chosen. The current selections are Do Not Display (no data shown); Decimal Degrees (mouse position in plane units); Deg, Min, Sec (mouse position in DMS notation for latitude/longitude); and Pixel Position (mouse position in fractional pixel units in the map).

Show Mouse At Cursor indicates that the mouse position and height will be shown next to the mouse cursor in the main display area as well as in the status bar.

Undo Levels is the number of levels of undo allowed in the program. The maximum number of undo levels is 32. Each undo level takes as much memory as the current data in memory, so using very many undo levels with a large surface will take an enormous amount of memory. You must always have at least one undo level.

### 3.2.5. Geoid Parameters

For the Surface→Point Process→Fit Geoid command (as well as some others dealing with a round world), a description of the world is required. The Geoid Parameters dialog allows the model to be adjusted for specific purposes. Usually, adjustment of these parameters is not required.




Model is the name of the geoid model to use. Some of the more popular ones are NAD-27 and WGS-84. A User model is provided which allows the individual parameters to be entered.

a, b, e12, and e22 are the geoid parameters. The data in these windows cannot be entered unless the model is set to "User Model". Unless you understand the meanings of these values, do not use the user model geoid. I will tell you that a and b are sphere radii and e12 and e22 are e1 and e2 squared, which are eccentricity measures. Using a equal to b and e12 equal to e22 which is equal to 0.0 should give a perfect sphere.

### 3.2.6. Flush Abort Command

Sometimes, when an action is canceled, the abort command doesn't properly reset. Every command that can be canceled after that immediately returns without doing much of anything. To fix that problem, use the Flush Abort Command menu item. It clears the problem and allows things to continue smoothly.

## *3.3. Surface*

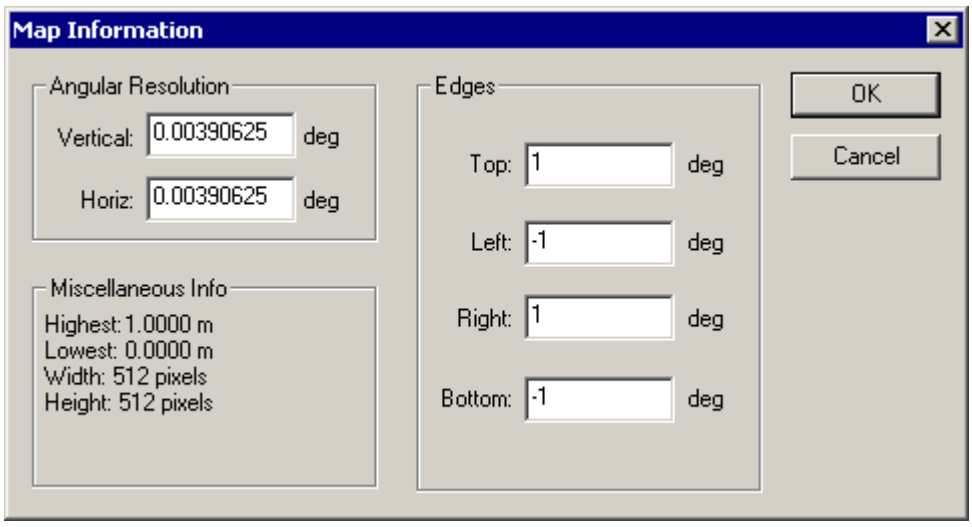
The surface menu contains the system options that deal directly with the surface data. Changes in the surface data will be reflected in the texture map when the map is recomputed (typically after each operation).

### 3.3.1. Find Min/Max

Locates the minimum and maximum altitude surface features and displays them in Map Information dialog (described next). Under certain circumstances, the system will lose track of the edges of the surface and/or the min/max values. Selecting this option will usually reset these parameters to appropriate values.

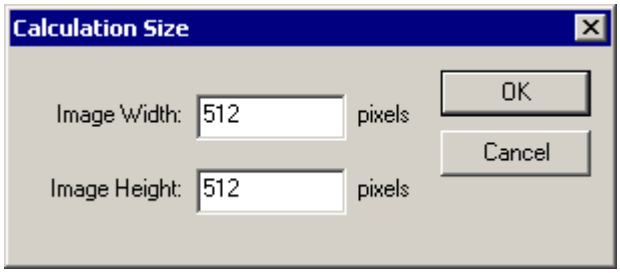
### 3.3.2. Map Info

Shows the current system information (such as what the edges of the map are, as well as the current vertical and horizontal resolution, what the system found to be the highest and lowest points the last time it checked, and the current surface size. Only changes made to the edges values will have any effect.




### 3.3.3. Size

The Size dialog lets you change the size of the surface. It is shown below. When a surface size is changed, all data in the surface is lost. The way to crop an image to a different size/location is to use the Resample command.

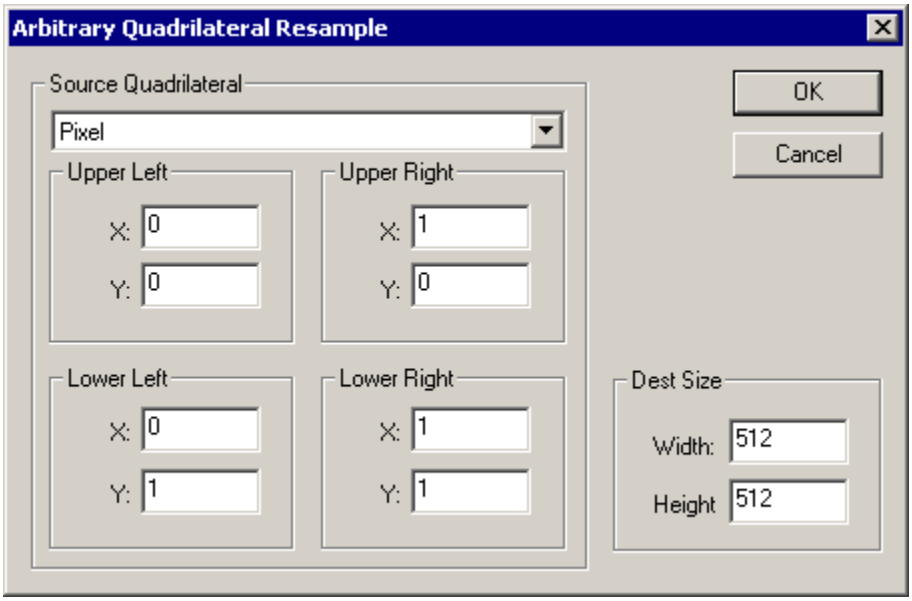



### 3.3.4. Resample

Resampling allows you to change the resolution and size of the surface.

#### 3.3.4.1. Quadrilateral Resample

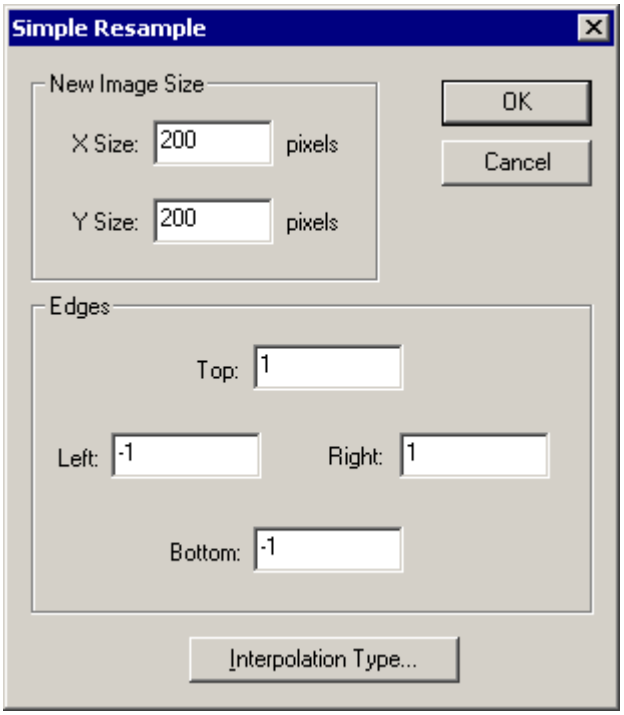
The Quad resample is very similar to the regular resample (see below) in that it allows portions of the existing terrain to be enlarged or reduced. Where it is very different, however, is that the area does not have to be rectangular. Rectangular operations can be performed using this command, but require extra care to ensure that the source quadrilateral is rectangular. The dialog used by this command is shown below.




The drop list just under ***Source Quadrilateral*** specifies the coordinate system of the quad: *Pixel* uses a pixel-sample based coordinate system, while *Plane Units* uses a lat/lon format for each coordinate in the local plane coordinates. ***Dest Size*** is the size of the new surface that will be generated, in pixels.

### 3.3.4.2. Simple Resample

Changes the resolution of the surface. Can be used to sample a large image down or a smaller one up. The type of interpolation can be chosen (closest-neighbor, linear, interpolating spline, approximating spline). It can also be used to crop an existing surface.



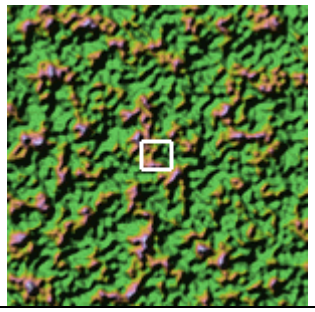
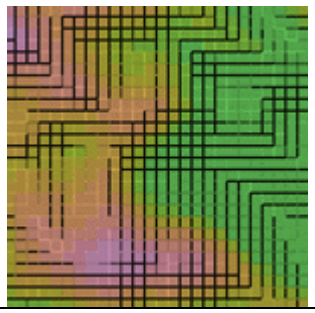
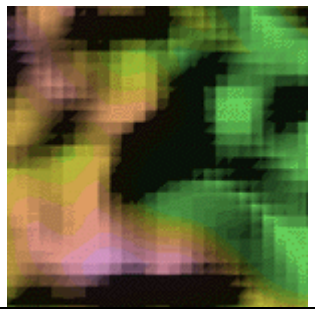
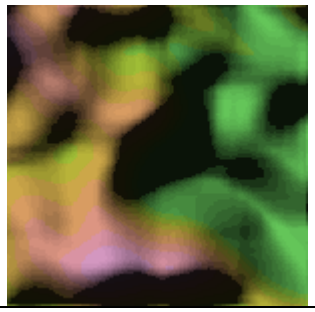

***New Image Size*** (**X Size** and **Y Size**) determines the size of the new image, in samples. The default values are always the current image size.

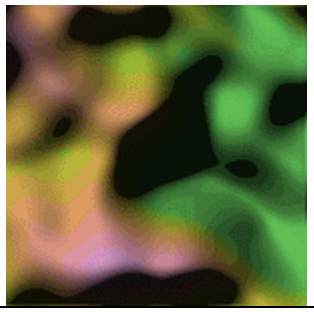
***Edges*** determines the area of the old image (in evaluation units, which are the ones that are displayed in the status bar when you move the cursor over the image) that will be resampled into the new surface. Any values can be entered here, but the defaults are always the edges of the current surface. If one or more edges of the source area are outside of the current surface, then that portion of the surface will be filled with a value of 0.

The system will make a new blank surface in memory and then copy the relevant portions of the current surface to the new one, and finally, dispose of the old one. What this means is that the amount of memory consumed may be very large. For example a 1024x1024 sample surface being resampled to a 2048x2048 surface will require about 35 MB (1K*1K*5+2K*2K*5) during processing.

The Interpolation scheme button brings up the system dialog that enables the type of interpolation.

The following set of images shows the difference among the different types of interpolation:

| Interpolation Type | Image | Description |
|---|---|---|
| Basis image |  | This is the reference image for the interpolation demonstrations. The white square is approximately the area that will be interpolated. |
| Nearest Neighbor |  | Shows the problem with nearest-neighbor sampling: the pixels form rectangular flat regions. This is the classic "mosaic" effect hated by lots of folks.<br>The unusual "box" coloration is caused by the sharp edges between elevation steps interacting with the lighting settings. |
| Linear |  | Interpolates from one corner to another. This interpolation gives a nice, faceted surface. |
| Spline Interpolation |  | Spline interpolation gives a nice smooth curve through the actual data points.<br>The spline used here is a Catmull-Rom spline.<br>Comparing this image with the approximating spline below, it is apparent that the interpolating spline leaves some unusual ridges and striations in the image. |

| | | |
|---|---|---|
| B-Spline Approximation |  | Spline approximation doesn't interpolate exactly between points, but can be outside the range of the original data set.<br>We use a B-spline for this program.<br>Comparing with the interpolating spline, this image is much smoother because it doesn't have to exactly match the surface points, but can merely approximate them. |
| Beta Spline Approximation | No image is shown because a beta-spline is basically just a cross between an interpolating spline and a linear spline. | A beta-spline is like a cross between a linear and spline interpolation. As the tension goes from 0 to infinity (or a very large number), the surface goes from the fully smooth b-spline type surface to the faceted linear surface. |

## 3.3.5. Rotate

### 3.3.5.1. Rotate 180 Degrees

Rotates the current surface 180 degrees clockwise (actually, it flips the surface around a diagonal line going from lower left to upper right). Performing the operation again rotates the surface back to its original orientation.

### 3.3.5.2. Rotate 90 Degrees Counterclockwise

Rotates the surface 90 degrees counterclockwise. If the surface is not square then the shape of the surface will be affected as well.

### 3.3.5.3. Rotate 90 Degrees Clockwise

Rotates the surface 90 degrees counterclockwise. If the surface is not square then the shape of the surface will be affected as well.

### 3.3.5.4. Flip Horizontally

Flips the image left-to-right, as if the image were being viewed in a mirror placed at the side of the image.

### 3.3.5.5. Flip Vertically

Flips the image top-to-bottom, as if the image were being viewed in a mirror placed at the bottom of the image.

## 3.3.6. Custom Paint Tool

The paint operation gives you a way to directly modify the surface data using the mouse pointer to manipulate the data. The controls are a little funny, but not *too* bad. The Paint menu option toggles on and off. When turned on, it first brings up the brush editing dialog (see Edit Paintbrush below) and click-drag-release sequences perform the painting operation on the relevant parts of the surface then recompute the texture map for the affected area.

Note that the paintbrush operates one mouse move at a time; moving the mouse over an area will cause overlapping areas to be processed multiple times. This action is not usually a problem, but can be in certain circumstances.

## 3.3.7. Custom Clone Tool

The clone operation copies part of the image from one place to another using the currently selected brush. The general cloning sequence is:

a) start the clone operation via menu or toolbar button,

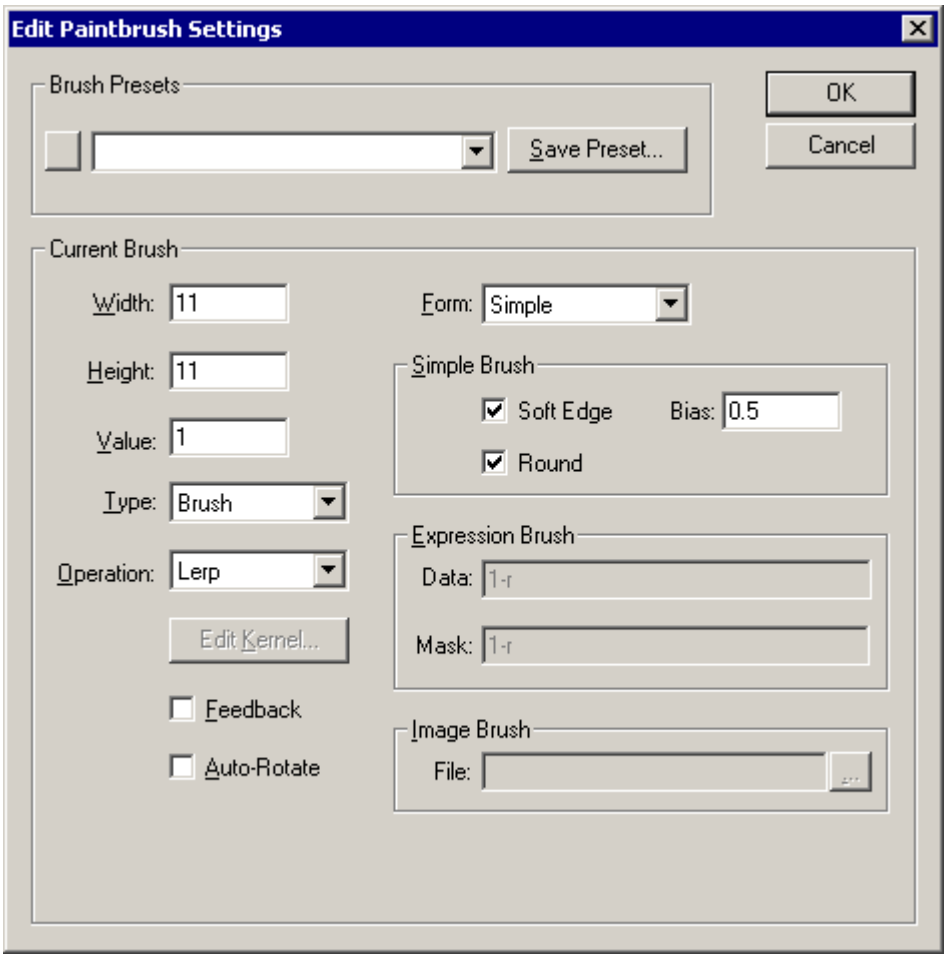
b) set the brush information in the Brush Setup dialog (see Edit Paintbrush, below),

c) select a data source by holding down the shift key, moving the mouse to the desired source, and clicking and releasing the left mouse button, and,

d) move to the desired destination and depress the left mouse button. Move the mouse around to copy the data from the surface, releasing the button when done. Pressing the button and moving the mouse again will make another copy of the data, centered at the new mouse point. Gives a quick and easy way to make multiple duplicates of an area of the surface.

### 3.3.8. Edit Paintbrush

The Custom Paint Tool and Custom Clone tool rely on a paintbrush to perform their actions. The settings on this paintbrush are controlled via the Brush Setup dialog, shown below.




The **Brush Presets** group provides an interface to manage brush presets. Wilbur doesn't ship with presets; you'll have to make your own. The unlabeled button to the left of the drop list changes the directory where the presets will be read. The drop list shows all brush preset files in the selected directory. Selecting an item from this list will load the file into the dialog. The **Save Preset** button provides a file save dialog that allows you to save the currently-specified brush settings to a new preset file.

**Width** is the width of the brush and **Height** is the height of the brush, in data samples. All values will be rounded up to the nearest odd number due to a quirk of Wilbur's internal brush processing code.

**Value** is the value used for the brush. What this values means depends on the exact operation being performed.

**Type** is the primary data source. The values supported are described below:

| Item in List | Description of data source |
|---|---|
| Brush | the brush profile |
| Convolve | a copy of the surface convolved with the paintbrush kernel. |
| Uniform Noise | the random number generator, generating uniformly distributed noise |
| Gaussian Noise | the random number generator, generating gaussion distributed noise |

If the *Convolve* brush type is chosen, the **Edit Kernel** button becomes available. Clicking this button brings up the kernel editing dialog (described in 3.6.9.3) for the kernel that will used during processing. Using a convolution brush will result in strange artifacts appearing when you hit the edge of the surface.

**Operation** specifies how the brush is combined with the surface. The supported operations (surface is the current surface value, source is the source determined by the Type option, and value is the value entered in the dialog) are:

| Item in List | Operation performed if |
|---|---|
| Lerp | Performs a linear interpolation between the current surface value and Value using the current source value: surface = lerp(source, surface, value) |
| Add | Surface = surface + source * value |
| Subtract | Surface = surface - source * value |
| Multiply | Surface = surface * source * value |
| Divide | Surface = surface / (source * value) |
| Minimum | (result * value) if (result * value) < surface |
| Maximum | (result * value) if (result * value) > surface |
| Exponent | Raises the surface to a the power of result * surface.<br>Surface = pow(surface, result * value) |

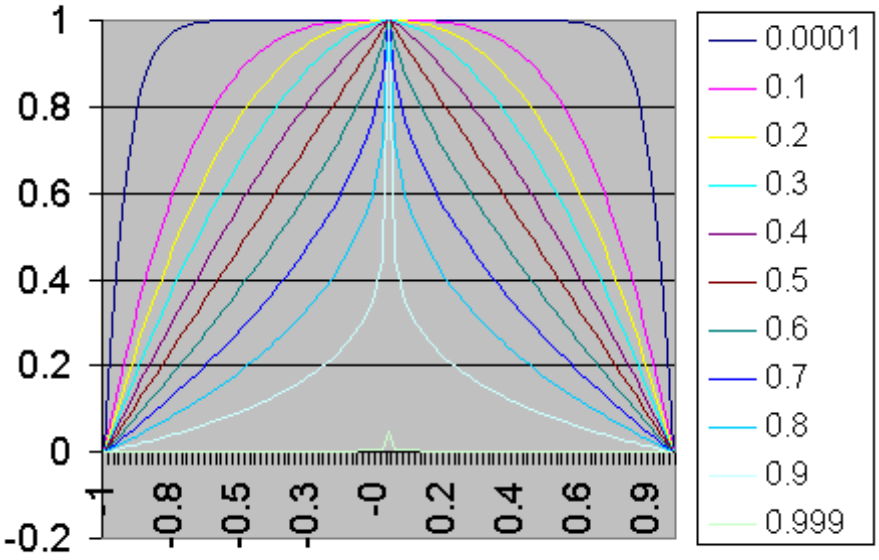
Yes, the brush operations are probably a whole lot more complicated than they need to be. On the other hand, you can do *lots* of stuff with the brushes.

**Feedback** indicates if the operation should work from the current surface or from the first undo level. As an example, enabling feedback with a blur brush will cause the blur to occur multiple times at a point as you drag the mouse back and forth over that point. Leaving feedback off in that case would result in a single blur being performed, regardless of how many times you stroke over the same point in the surface.

**Auto-Rotate** indicates if the brush should be rotated in the direction of mouse travel. When checked, the brush will rotate along the path of travel, but will take slightly longer to draw.

**Form** is the type of brush that will be used. *Simple* brushes provide just a little control over the brush effect. *Expression* brushes let you define an expression to place data into the surface and another to mask the data. *Image* Brushes load an image from disk as use that as a stamp for the brush operation (image brushes do no have a mask).

The Simple Brush group are the features used when Form is set to Simple. Simple brushes are round or square (or elliptical / rectangular if you don’t set the width and height to the same number). **Bias** is the value used to compute the brush profile for a simple brush. It effectively controls the "softness" of the brush. A graph of this function for various values is shown below. The horizontal range from -1 to +1 is scaled from -Radius to +Radius. Note that if the simple brush is not indicated as *Soft Edge*, the effect is the same as a bias value of 0.




The Expression Brush group are the options needed when Form is set to Expression. The two expressions used for the brush are defined on the range of -1 to +1 in X and Y (regardless of brush size). An expression brush that replicates a round, soft-edged simple brush with a 0.5 bias can be replicated by placing "1-r" into the Data expression and "1-r" into the mask expression. All of the functions described in the Calculate Surface section are available in these expressions. (A simple soft brush with arbitrary bias can be generated by using "bias(b, 1-r)" as the Data expression).

The Image Brush group provides you a location to enter the name of an image to use as the brush. This image will be distorted to fit the Width and Height of the brushes.

In addition to the dialog described above, all of the paintbrush tools also provide a tool options toolbar for the simpler controls such as size. This toolbar is described below:


**Width** is the width of the tool, in data points. This value can also be set in general terms using the green controls on the painting toolbar.

**Height** is the height of the tool, in data points. This value can be set in general terms using the green controls on the painting toolbar.

**Value** is the strength of the tool.

**Operation** is the type of operation that will be performed by the brush tool.

**Auto-Rotate** indicates if the brush should be rotated in the direction of mouse travel. When checked, the brush will rotate along the path of travel, but will take slightly longer to draw.

**…** selects the preset directory. The **Presets** drop list provides a list of presets in the current directory. Selecting an item from this list loads the brush preset. **Save…** saves the current brush settings as a brush preset in the current brush presets directory.

The other options described in the full dialog are not present on the toolbar.

### 3.3.9. Flood Fill Tool

The floodfill tool lets you pick a point on the surface and have the system perform a floodfill operation from that point. The options for the floodfill tool are shown on the tool options toolbar and are described below:


The floodfill operation will be performed starting from the current mouse position and will affect all contiguous samples between High and Low. New Value will be the value that replaces those points.

### 3.3.10. Gradient Tool

The gradient tool will draw a gradient from the start point to the end point of a line that you draw by clicking and dragging. The gradient parameters are specified in the gradient tool options toolbar as shown below:

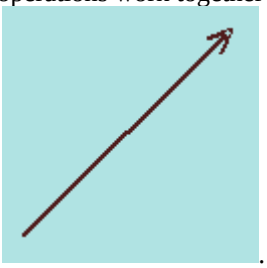

**Type** indicates the type of gradient that can be drawn. The types are shown in the table below.

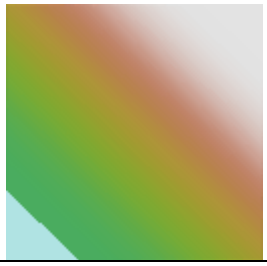
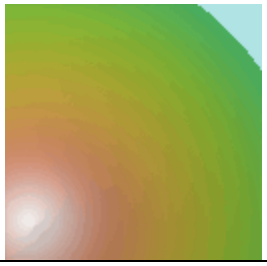
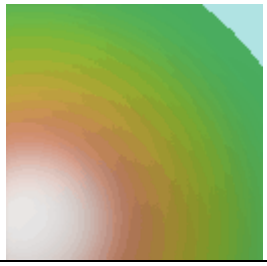
**Low Value** is the value of the starting point on the gradient.

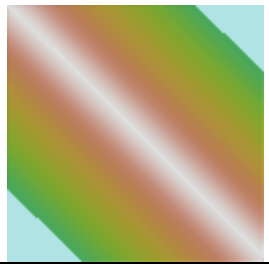
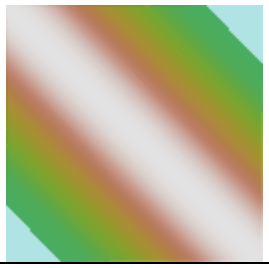
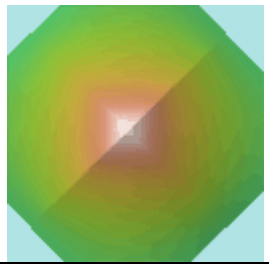
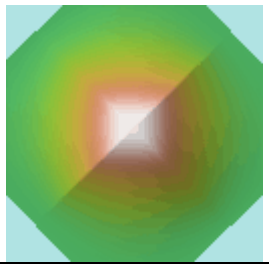
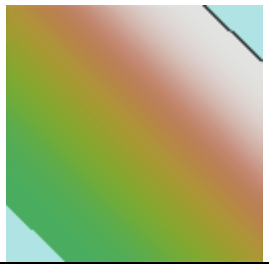
**High Value** is the value of the ending point on the gradient.

**Blending** is the blending function from the low point to the high point. Linear is a straight line blend (has sharp bends at top and bottom), while Cosine is a cosine-based blending, which smoothly joins at top and bottom of the gradient.

**Operation** indicates how the gradient will be blended with the existing surface.

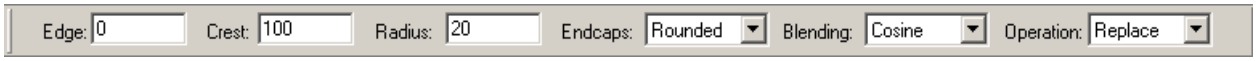
The table below shows how the type and blending operations work together to generate a gradient from a simple swipe drawn from lower left to upper right like so:  .

| | Blending: Linear | Blending: Cosine |
|---|---|---|
| Type: Linear |  |  |
| Type: Radial |  |  |

| | | |
|---|---|---|
| Type: Double |  |  |
| Type: Pyramid |  |  |
| Type: Ramp |  |  |

## 3.3.11. Paint Line Tool

The line tool draws a line in a manner similar to the gradient tool above. Select the tool, click, drag, and release to draw a line. The options for the line painting are specified on the tool options toolbar as shown below:




**Edge** is the value used at the outermost edges of the drawn line.

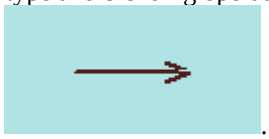
**Crest** is the value used along the center of the drawn line.

**Radius** is the distance from the edge to the center (half the total width).

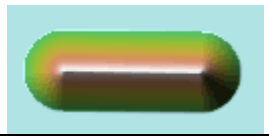
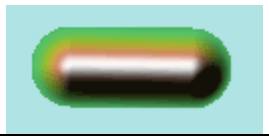
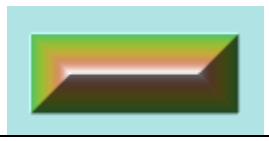
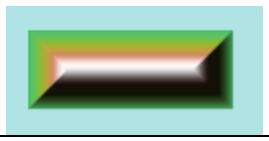
**Endcaps** are the types of endcaps that can be drawn on the line. See the table below for examples of how the different endcaps appear.

**Blending** is the blending function from the low point to the high point. Linear is a straight line blend (has sharp bends at top and bottom), while Cosine is a cosine-based blending, which smoothly joins at top and bottom of the gradient.

**Operation** indicates how the gradient will be blended with the existing surface.

The table below shows how the type and blending operations work together to generate a line from a simple swipe drawn from left to right like so:  .

| | Blending: Linear | Blending: Cosine |
|---|---|---|
| Endcaps: None |  |  |

| | | |
|---|---|---|
| Endcaps: Rounded |  |  |
| Endcaps: Square |  |  |

## *3.4. Texture*

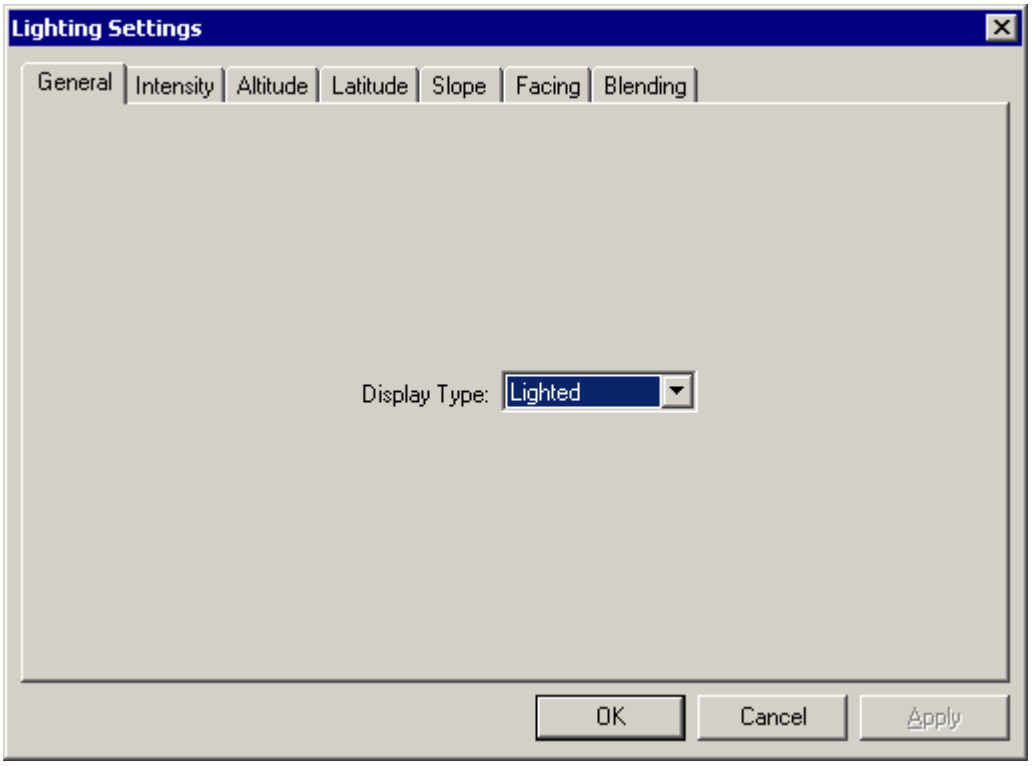
The entries on the texture menu deal purely with the appearance of the texture map (the pretty picture) and not with the underlying data set. There is an operation that will transfer the texture map to the surface to allow you to play with it, though.

### 3.4.1. Lighting

The Lighting menu option provides access to the texture map rendering facility. These pages are General, Intensity, Altitude, Latitude, Slope, Facing, and Blending.

#### 3.4.1.1. General Page

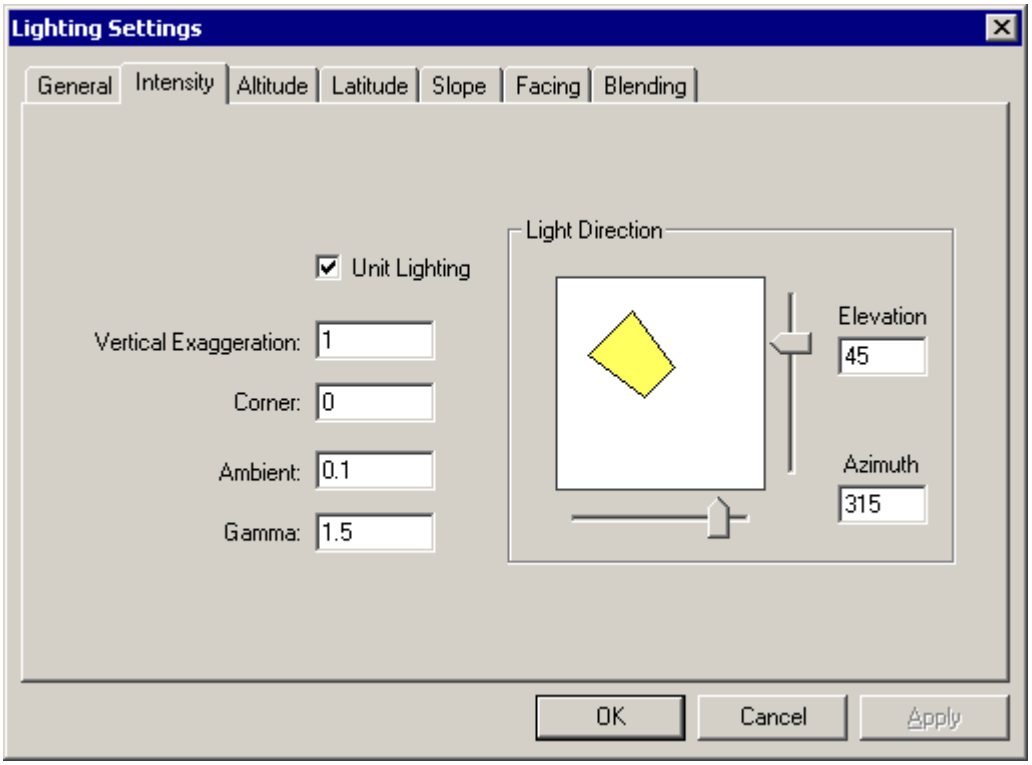
This page is used to determine, in the broadest sense, the appearance of the computed texture map. It is shown below.




The **Display Type** drop list selects the type of map that will be computed. The *Lighted* option indicates that the surface will appear as if lighted by a light source (described on the intensity page), while the *Height Code* option just colors the surface without any lighting.

### 3.4.1.2. Intensity Page

The Intensity page controls the appearance of lighting on the surface. If the Display Type control on the General page is set to "Lighted", this page has meaning; if the Display Type control is set to "Height Code", then the contents of this page are ignored.




**Unit Lighting** indicates if the lighting model should attempt to take into account the real distance between pixels in the surface map when calculating lighting (Unit Lighting unchecked) or just assume that the distance between pixels is 1 (unit Lighting is checked). Usually, unit lighting gives good results, but may not be completely accurate in some situations.

**Vertical Exaggeration** is the amount by which the vertical dimension of the surface is stretched during the lighting calculations. A value of 1 indicates that the correct altitudes are used in the lighting calculations. As the value increases from 1, the surface will appear to get steeper and steeper. Similarly, as this value is reduced below 1, the surface appears to get flatter and flatter.

**Corner** provides a way to weight the corner pixels (not just the up/down, right/left pixels) for the current pixel being lighted. Usually, the effect that this has is to blur the appearance of the surface. This option may be desirable for certain types of surface, such as plasma surfaces, which have very strong axis alignments.

**Ambient** controls the color that appears in the darkest area of the intensity calculations. Without this parameter, those areas facing away from the light would appear completely black. The value ranges from 0.0 (completely black when facing away) to 1.0 (completely lighted even when facing away).

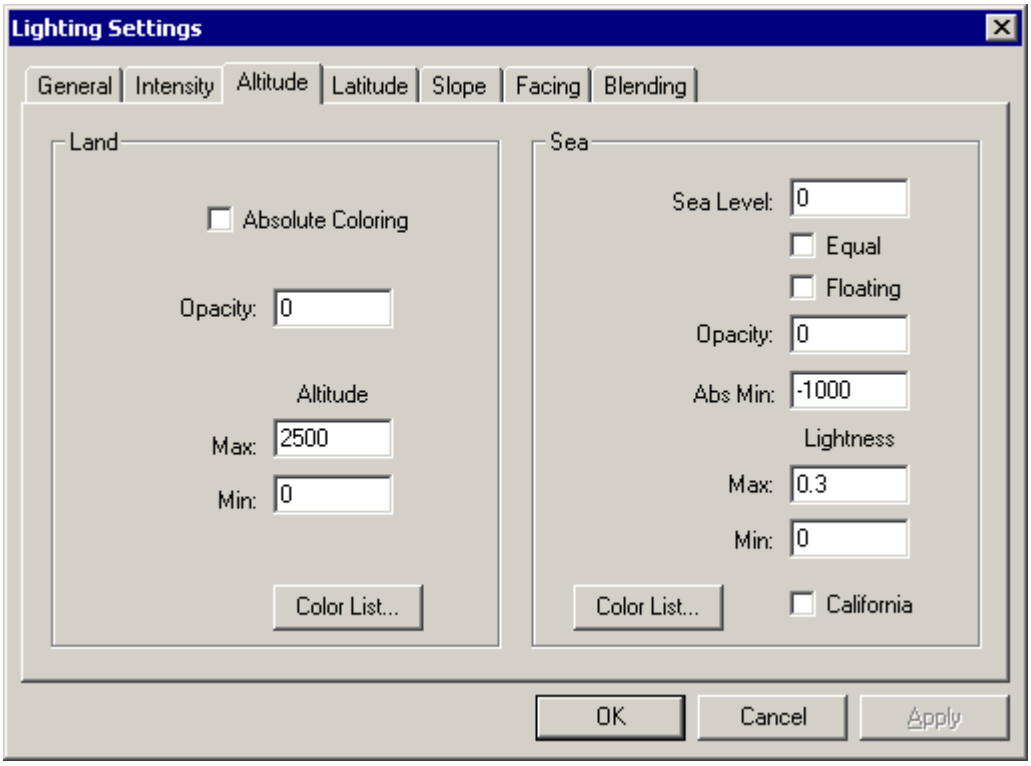
**Gamma** controls the intensity curve used during the calculations. This parameter can be used to weight differing intensities to account for differing display adapters as well as for aesthetic effects.

**Elevation** is the angle above the horizon at which the light appears. This value is in degrees, with 0 degrees being on the horizon and 90 degrees being directly overhead.

**Azimuth** is the angle around (north is 0 degrees, degrees are measured clockwise) at which the light appears.

### 3.4.1.3. Altitude Page

The altitude page controls the coloring of the surface according to altitude.




**Absolute Coloring** indicates if the coloring from the color list should be based on the highest and lowest points on the map (if unchecked) or on the min/max altitude values specified (if checked).

**Min** and **Max Altitude** are the minimum and maximum altitude, respectively, that the color list will map to. Values above sea level and at or below **Min** will map to color value 0 in the color list; altitude values at or above **Max** will match to color list value 1.0.

**Sea Level** is the altitude below which the sea coloration takes effect. If no parts of the map are below sea level, no sea coloration information will be used. Setting **Sea Level** to a large negative value (-100000 for instance) will ensure that only the land colors are used.

**Equal** indicates if the sea coloration is to be applied to all values below sea level (if unchecked) or only to those values exactly equal to sea level (if checked). Some data sets have sea level as 0, which causes any land areas below sea level (like the Dead Sea area in Israel or Death Valley in the United States) to be colored as sea. Setting this flag to true will only color those pixels exactly equal to sea level, causing only some pixel to be colored, not the whole undersea area.

**Floating** indicates if the altitude coloring should be calculated relative to the absolute height (if unchecked) or relative to sea level (if checked). For example, say that sea level has been set to 1000. With the floating value checked, sea level will be treated as altitude 0 for coloring purposes; if unchecked, it will be treated as altitude 1000.

**Opacity** indicates the amount of the intensity parameter that feeds through. Its default value is 0, which means that all of the intensity information in the ocean basin feeds through. To get rid of the sea information but preserve the depth coloring, set this value to a large negative number, like -100000.

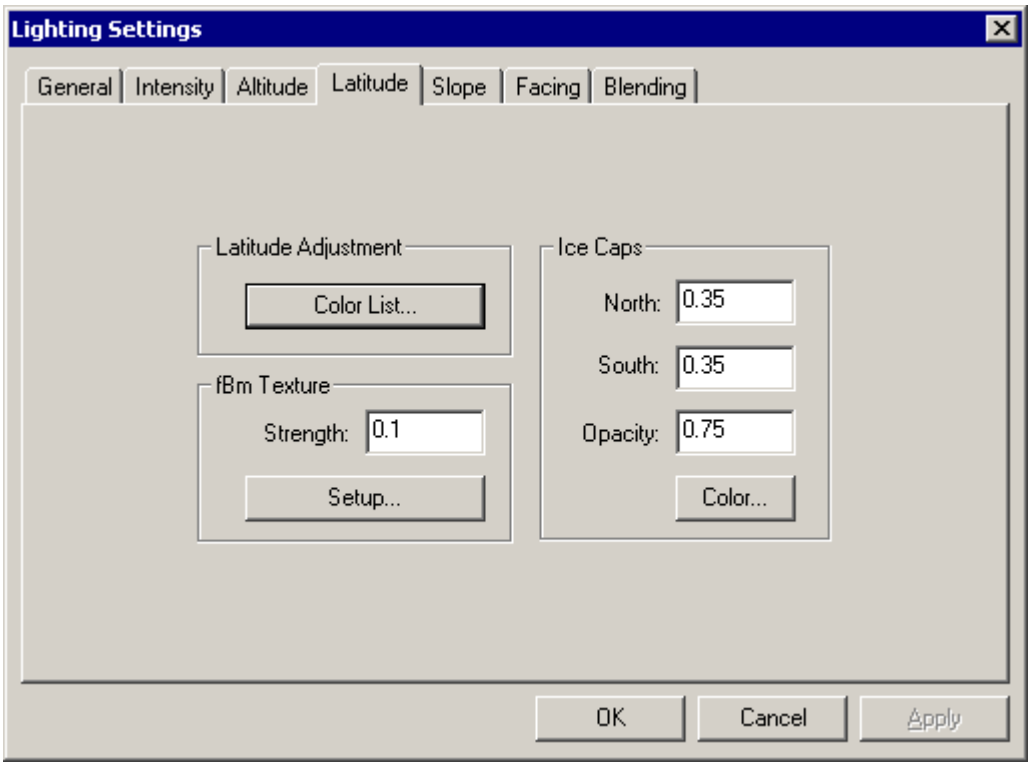
**Abs Min** is the absolute minimum value of the sea coloring. This value will correspond to the greatest value in the color list.

**Min** and **Max Lightness** control an auxiliary parameter for sea information. Oceans tend to get lighter as they get shallower, not just change color. To simulate this, the values entered here are added to the RGB component of the ocean color according to the sea depth.

The **Color List** buttons allow for the land and sea color lists, respectively, to be edited.

### 3.4.1.4. Latitude Page

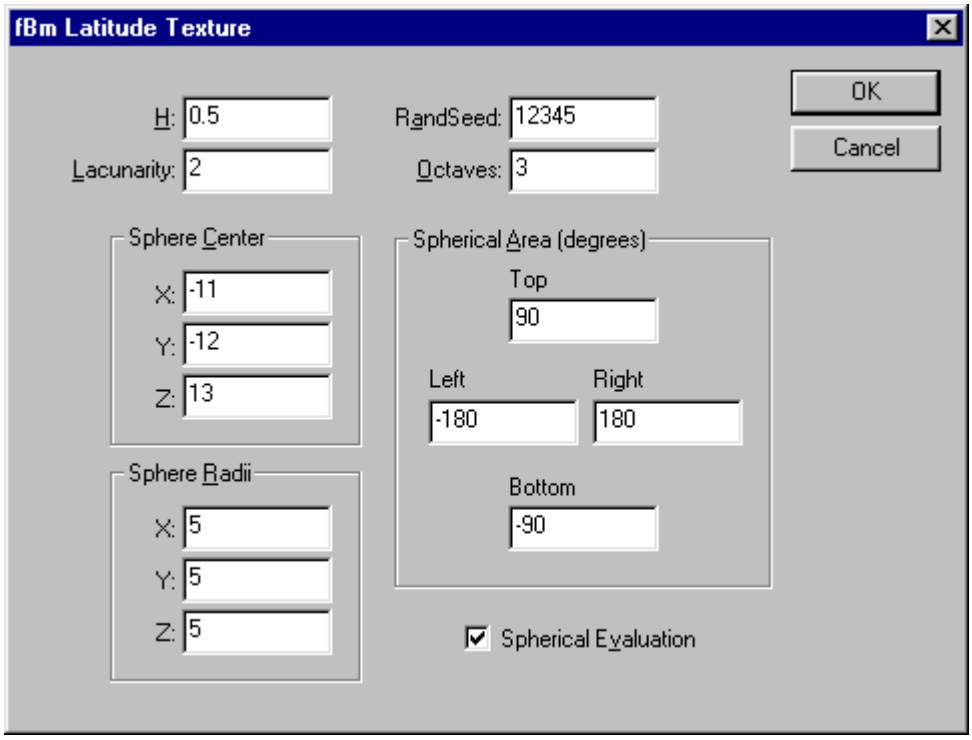
The Latitude Page controls the coloration of the surface with respect to latitude (the distance from the equator to the poles). The page is shown below.




***Latitude Adjustment*** **Color List** allows the coloring with latitude to be directly specified.

***Ice Caps*** describes the ice caps shown on the map. **North** and **South** are the range above which the latitude effect (latitude + fBm) is ignored and the ice cap color is used instead (a value of 0.2 gives mostly ice caps, 0.9 mostly without ice caps). **Opacity** relates to how much of the lighting intensity shows through the ice caps. Small values (around 0) show all of the underlying terrain details; large values (around 10) obscure all traces of underlying terrain details, leaving only the ice cap color. Finally, the **Color** button sets the color to be used for ice caps.

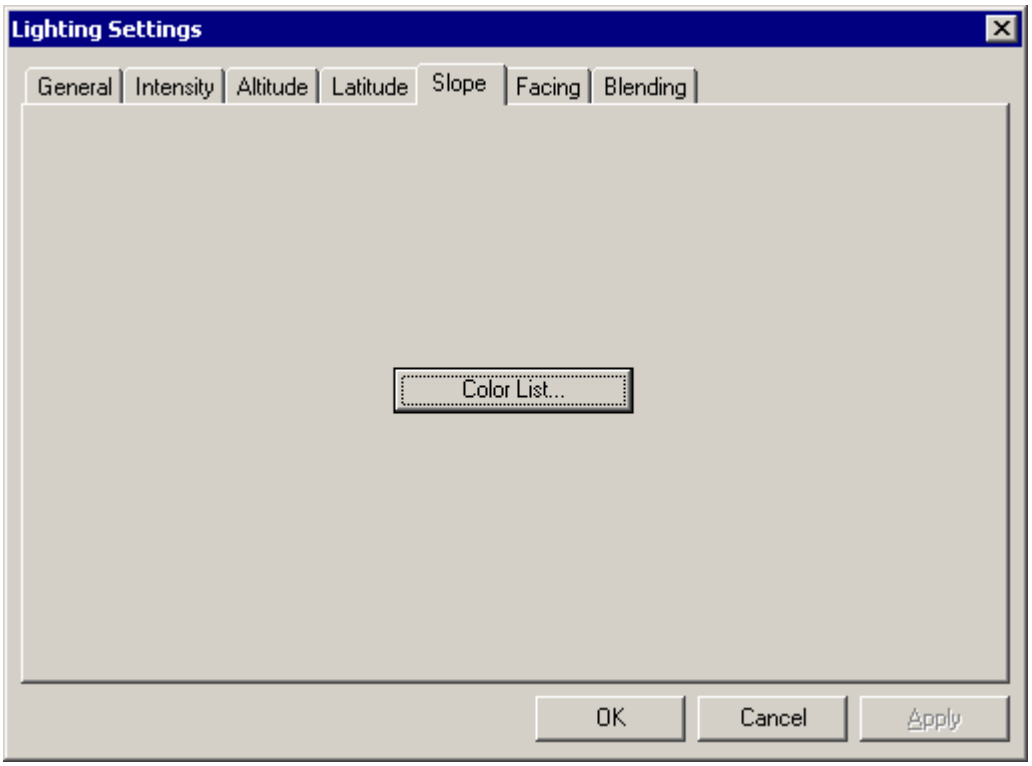
***fBm Texture*** controls the adjustment of the surface by an fBm texture that is added to the latitude value before being used as an index into the latitude color list. Because the base fBm value is in the range of -1 to +1, the strength parameter is used to modify this value. The effect can range from very subtle (small values) to overwhelming (large values). [Note: the current software version uses the fBm value as an addition to the latitude value, which is added to the altitude for a final index into the land color list.] Note that setting the fBm Texture Strength value to greater than 0 will substantially increase the calculation time required. In addition to the strength parameter, the Setup button calls the dialog shown below, which is used to control the appearance of the fBm noise used.




The parameters on this dialog are the same as those used for calculating a height field. See section 3.6.1 for more information.

### 3.4.1.5. Slope Page

The Slope page allows coloration to be changed according to the local slope of the terrain. This page is shown below.

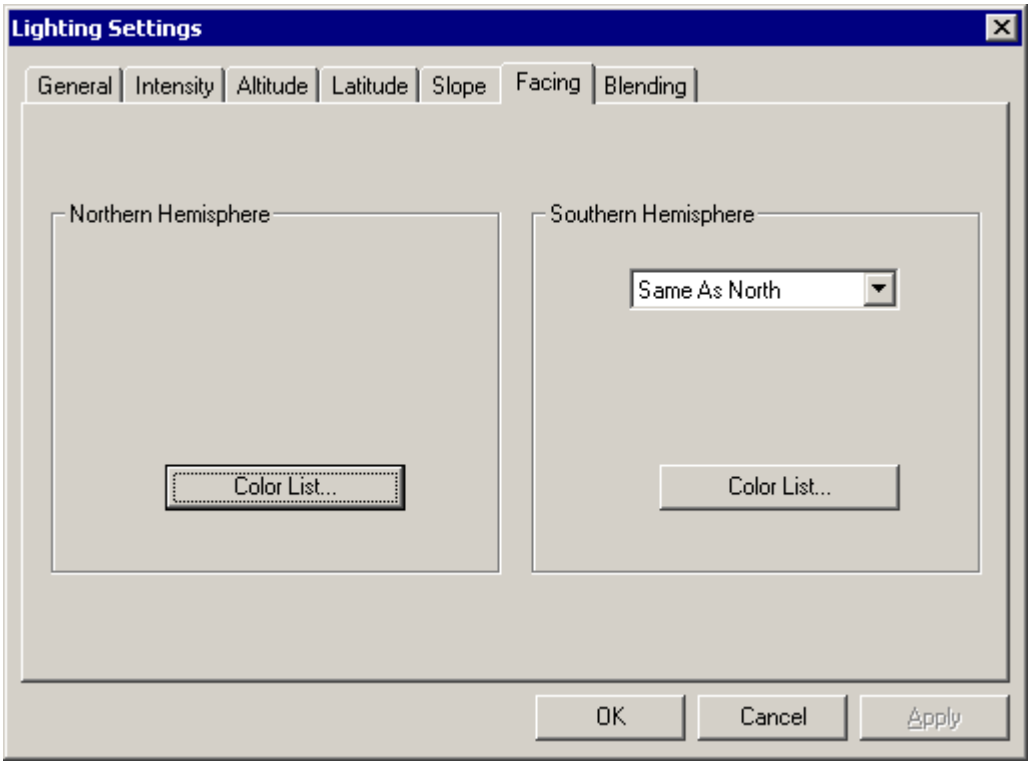


The Slope page controls how the coloration is affected by the slope of the land. For example, it may be desirable to have nearly flat areas be relatively greener than more nearly vertical ones. Similarly, it may be desirably to have nearly vertical ones colored a specific color (gray, for example, to represent cliffs).

The **Color List** button controls the color list used for the slope coloration.

### 3.4.1.6. Facing Page

The Facing page allows the facing of the terrain to have an effect on the final coloration appearance. This sort of coloration is found in the real world where, for example, the vegetation on the south facing side of a hill is different than that on the north facing side of the hill. The facing page is shown below.




The facing page is broken into two major parts: ***Northern Hemisphere*** and ***Southern Hemisphere***. This break is indicated by the fact that the slope of the earth is opposite in the northern and southern hemispheres.

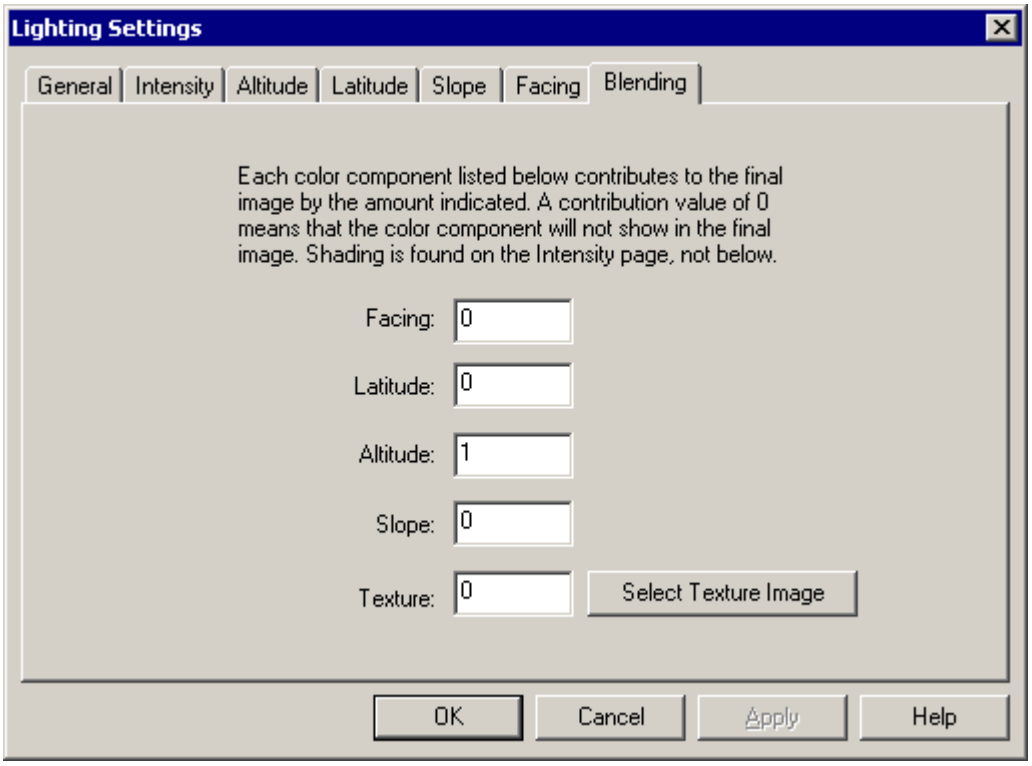
For both the northern and southern hemispheres, **Color List** controls the color list used.

***Southern Hemisphere*** has an extra control, one that determines its relationship to the northern hemisphere. Because the light direction (what we're assuming to be the cause of the coloration difference) in the southern hemisphere tends to be opposite that in the northern hemisphere, the control can set the coloring to the same as the northern hemisphere, opposite the northern hemisphere, or totally independent of the northern hemisphere.

As an example of how this page might be used, consider a case where south-facing slopes should be colored a tan color and north-facing slopes a dark greenish-brown (like the Sierra Nevada slopes in California). Create a color list with four colors: the darker color as the first color, the lighter color as the next two colors, and the original darker color as the fourth color. The value of 0.00 (color 1) will be mapped to north, with 0.33 being southeast (color 2), 0.66 (color 3) being southwest, and 1.0 (color 4) north again.

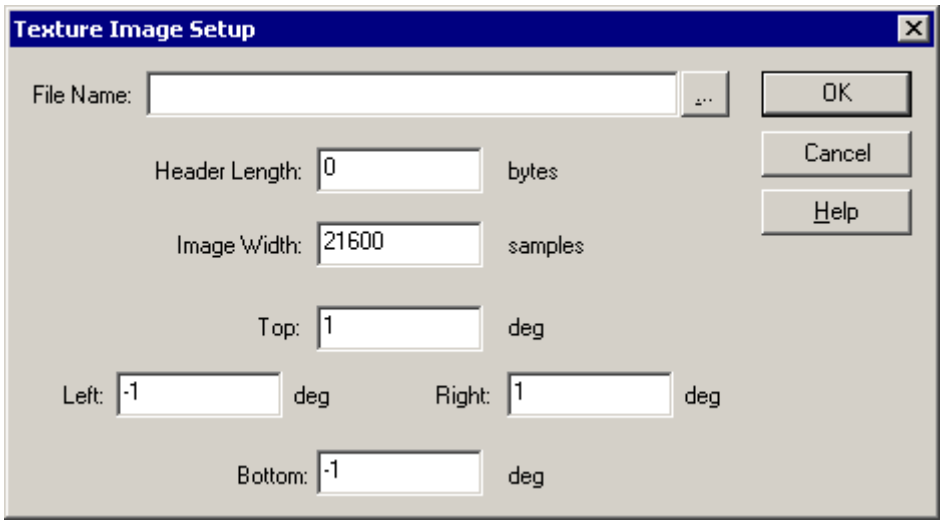
### 3.4.1.7. Blending Page

The blending page controls how the lighting components are combined into a final pixel. The page is shown below.




The numeric values control the relative proportions of the components that are mixed (simple linear mixing only, sorry). All of the values are summed and each component is divided by the total to given the proportion of each pixel to be blended. The color values generated by the Facing, Latitude, Altitude, and Slope are controlled by their respective pages. Note that the final component (the intensity) is not shown on this page. The intensity component does has a similar control, the Ambient lighting component, which can be found on the intensity page.

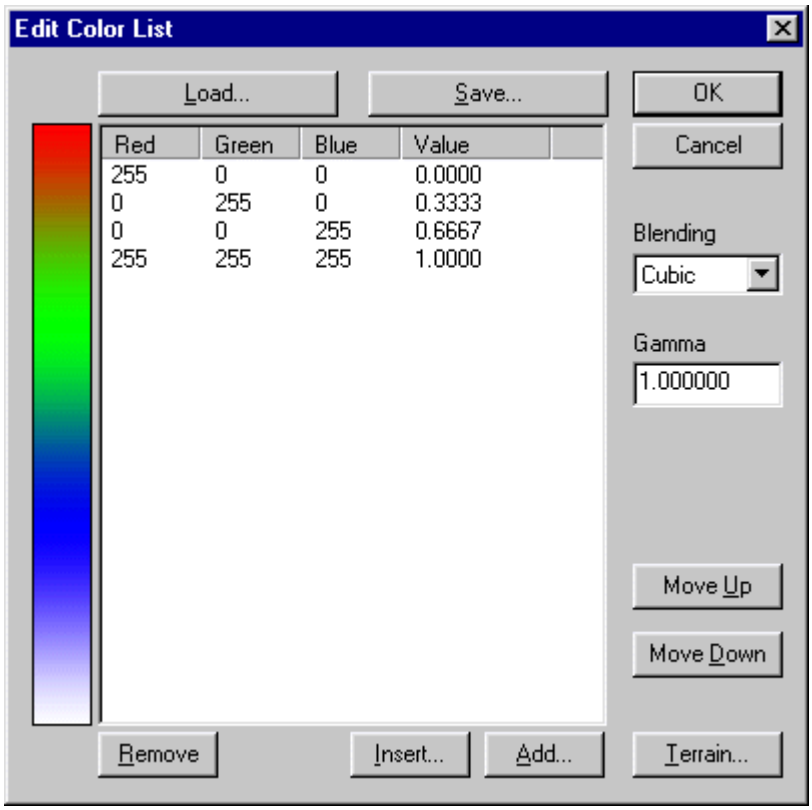
Texture allows you to select a texture image and use that as a color component as well. Clicking the Select Texture Image button brings up the dialog shown below:




This dialog allows you to select a raw RGB image and use it as a contributing color channel for the final image. I like to use it to map a big chunk of a relatively low-resolution Earth image and then apply the intensity from a higher-resolution section of terrain to allow me to generate the appearance of higher-resolution data even when that data isn't available. If you have selected a regular image type such as JPEG or BMP, then the Header Length and Image Width options will not be available.

### 3.4.1.8. Color List Edit Dialog

The color list edit dialog allows (as the name implies) the contents of a color list to be edited. It is shown below:




At the left edge of the dialog is a bar that shows the current color list. This bar is designed to work with a 16-bit or 24-bit video system and so may cause interesting effects with a simple 256-color display.

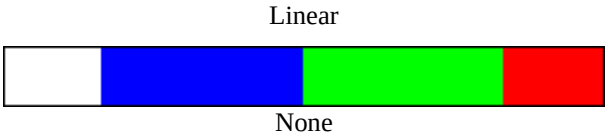
The list of values in the center of the screen is the RGB color values stored in the list, along with the parameter value that will represent that color in the final output. Note that the current software version does not allow the parameter value to be changed; it is computed automatically. To select a color, the Red component of that color must be selected (it's a problem with the Windows 95 list control).

The Load button will replace the contents of the color list with the contents of a color file from disk. The file format used is very simple: it is a text file with number separated by either spaces or commas. Each three numbers are taken to be an RGB color value with each component having a range of 0 to 255. This format is similar to that used by the Fractint program, except that this software does not allow for comments in the file. Carriage returns are also ignored.

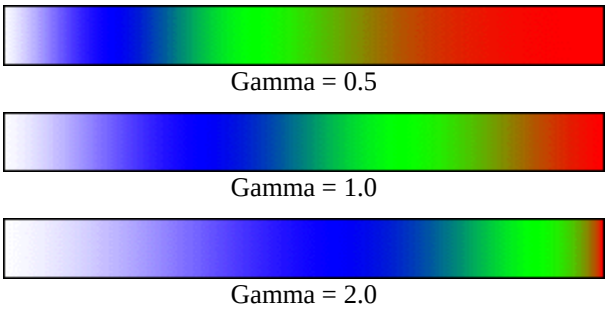
The Save button save the current color list data to disk as a text file with three numbers per line, each representing the RGB color value of the entry. The Blending and Gamma values are not saved.

The Blending function determines how the colors in the list are blended together. It has three values: none (the closest color in the list is used), linear (colors area linearly interpolated), and cubic (colors are interpolated using a Catmull-Rom spline). The following pictures show the differences between the type of interpolation:

Cubic




The "Gamma" value applies a non-linear scaling factor to the input value before it is used to index the color list. The following example shows how the gamma value can be used to affect the appearance of the above color lists (cubic blending selected):




**Move Up** and **Move Down** move the currently selected color item up or down through the list as indicated by the selected button.

**Insert** inserts a new color value into the list immediately before the currently selected color.

**Add** adds a new color to the end of the color list.

**Remove** removes the currently selected color from the list. If the shift key is depressed when this button is clicked, all entries in the list will be removed.

**Terrain** allows a color palette representing a terrain altitude map to be created. This map is the color map used by older software versions. The dialog invoked by this command requires three values: the number of colors, and the minimum and maximum parameter values. The parameter values used with the terrain color palette are shown below:

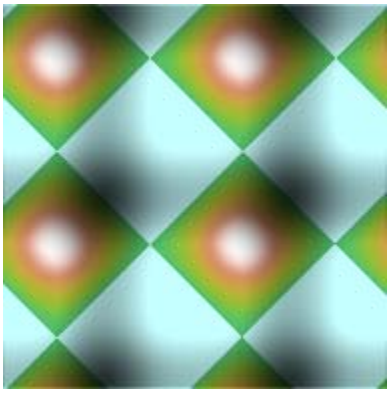

### 3.4.2. Recompute Lighting

This menu item forces the minimum and maximum of the surface to be recomputed and then the lighting model is applied to the surface.

### 3.4.3. Gray Maps

The Gray Maps menu item provides a home to the grayscale map options. Gray maps are provided for height information, elevation (slope) information, azimuth angle (phase) information, and a masking operation to make sea maps.

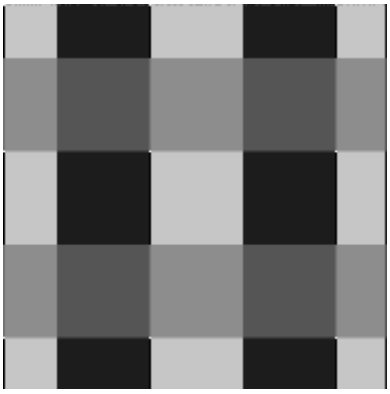
For the discussions below, a simple height map will be used. It is shown below. The height data was computed from the math function "sin(x)+sin(y)" calculated from -2*pi to +2*pi in X and Y.



Basic Lighted image using the default lighting operations.

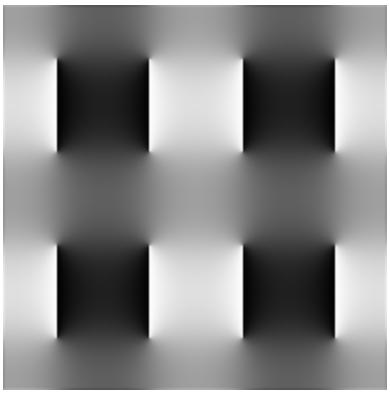
### 3.4.3.1. 8-way neighbors

This map represents the downhill flow from each pixel to one of its 8 neighbors. For the map used here it's not very exciting. It isn't very interesting even on regular terrain.



### 3.4.3.2. Azimuth (Phase) Map

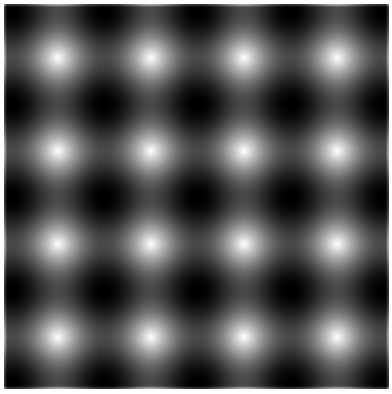
A phase map converts the texture map to a grayscale representation of the horizontal direction that each surface element is pointing. Value 0 is North, 63 is East, 128 is South, and 191 is West.



### 3.4.3.3. Elevation (Slope) Map

Provides a grayscale of the slope of the surface at each point. The intent was to scale the surface from 0 = 0 elevation (flat) to 255 = 90 deg elevation (vertical). The final scaling is somewhat off, however, so take the results with a grain of salt. After selecting this option, the software will ask "Adjust to full dynamic range?". If the answer is yes, the slopes will be rescaled to use the full 0-255 pixel range for coloring; otherwise, the coloring scheme

described above (which can be a little uninteresting). If the full range option is selected, a dialog will appear after the operation is complete that details the correspondence between pixel indices and slope.



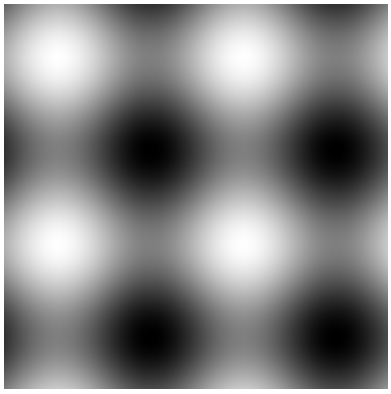
Scaled to full range



No scaling

### 3.4.3.4. Height Map

Calculates an image of the surface where index 0 in the palette is the lowest point on the map and value 255 is the highest point.

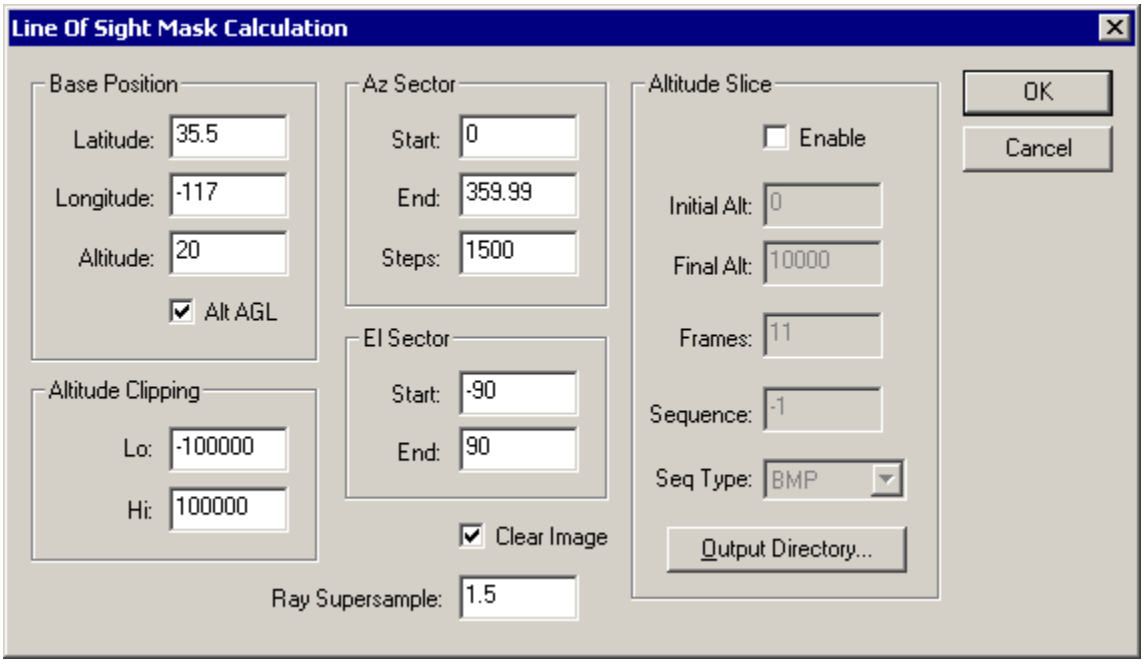


### 3.4.3.5. Light Map

The light map option is similar to the lighting option, but with two fairly simple differences: the direction of the light is the only parameter allowed to vary and the value coding doesn't stop at 90 degrees to the light direction. What this map generates then, is a a correlation between the lighting direction and the surface normal (the direction the surface is facing). If the surface is exactly facing the light, then the result will be white. If the surface is exactly facing away from the light, then the result will be black.

### 3.4.3.6. Line Of Sight Mask

The LOS Mask is a somewhat complex operation. What it calculates is the visibility of other points on the surface from the given point. If the point is visible, the output is white; if not visible, the output is black.




**Base Position** is the location from which the LOS mask will be calculated. If the Alt AGL button is checked, then the altitude value is relative to the surface height at that point.

***Altitude Clipping*** (**Lo** and **Hi**) values allows the mask evaluation to be constrained to a range of altitudes. These values do not affect the altitudes calculated, merely whether the final mask is marked or not.

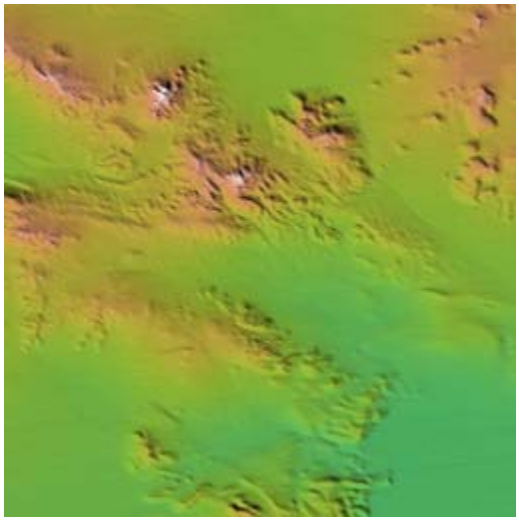
***Sector Scan*** allows only a part of the full circle to be evaluated. **Start** and **End** are the starting and ending azimuth values of the scan, in degrees. **Steps** is the number of times the LOS algorithm will be evaluated in that sector. For large maps (512x512) a value of 8000 or more may be needed here to ensure complete coverage.

**Ray SuperSample** is the frequency with which the surface is sampled. A value of 1.0 will sample exactly once per pixel. A value greater than 1 will give better results at the cost of increased run time.

**Clear Image** indicates if the image should be cleared before calculating. If it is checked, then the result will be a white mask on a black background. Otherwise, the result will be a gray mask on whatever the current map is.

***Altitude Slice*** allows the LOS evaluation to operate in a slightly different way. The default check is to determine what ground areas are visible from the selected point. The Altitude Slice algorithm checks if the given altitude is visible from the point. This feature can be useful for determining operational envelopes for ground-based systems, for example. If the enable box is checked, the altitude slice algorithm is used. For a total of **Frames** altitude slices starting at **Initial Alt** and going through **Final Alt**, the system will evaluate the LOS mask for the altitude. If the **Sequence** value is >= 0, the mask will be saved as either a BMP or PCX files, based on the **Seq Type** setting.

For the sample images, I loaded a terrain image near where I live, then computed a line of sight mask from the center of that image (35.5 lat, -117.5 lon, 200m AGL for those interested).



Base Image

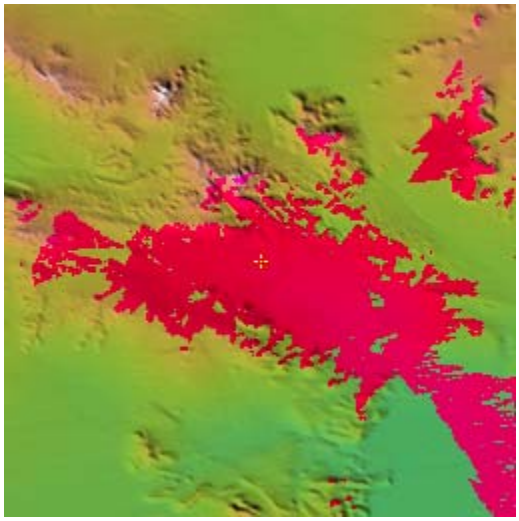
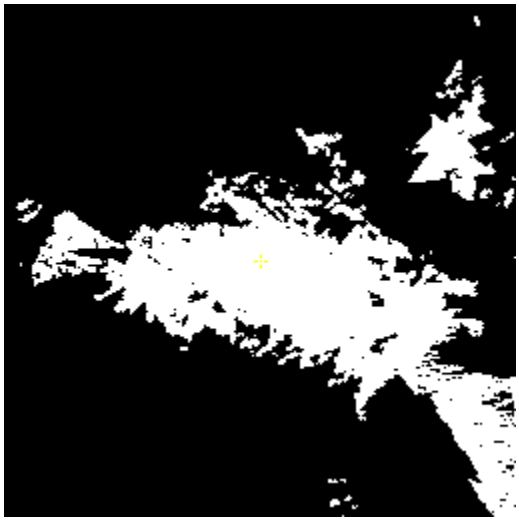


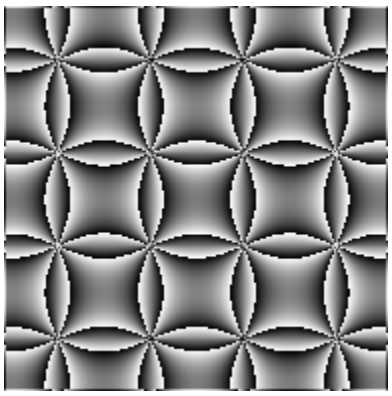
Image with mask (Clear Image not checked)



Straight image mask (Clear Image checked)

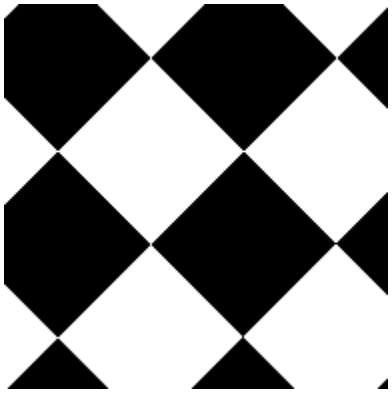
### 3.4.3.7. Quilt Map

During the development of the phase maps, I had a programming error that resulted in beautiful maps for certain types of functions. The best functions to use are nice, smooth functions, preferably with lots of transcendental functions (sin, cos, etc.) in them. The function takes two parameters: scale, which affects the number of iterations of "quilting" per azimuth block, and offset, which controls the symmetry of each block. Scale can vary to any desired value; offset is best kept within the range 0 to 1, where its effect is vaguely similar to the "bias" parameter in the surface paintbrush.



### 3.4.3.8. Sea Mask

There are instances in map preparation where it is necessary to calculate separate maps for land and sea and then merge the two. For example, in the maps of North Africa near the Nile delta, there is an area that is well below sea level on the same map as areas where there is ocean. The Caspian sea, Death Valley, places in Australia, and other areas around the globe present similar problems. My solution has been to calculate one map for land with sea level very low, one map for the sea, and one sea mask. The sea mask will be white (or black, if desired) over the areas below sea level and the opposite color above. Using a painting program such as Photoshop, the offending areas of the sea mask can be painted out. The repaired sea mask can then be used to merge the land and sea maps in Photshop.



## 3.4.4. Draw

The draw pop-up menu contains operations that don't really deal well with the surface but are kinda utility functions.

### 3.4.4.1. Clear Edges

Clears the edgemost pixels of the image to black. Makes a quickie frame.

### 3.4.4.2. Fluff Edges

Fluff copies the second row of pixels in from each edge to the edge itself.. This operation has the effect of putting up a row of pixels that looks "close" to the edge and will make seams less obvious if the map is tiled or wrapped onto a sphere. This operation isn't really required any more because of improvements in the lighting algorithm which removed most of the black lines along edges, but it's left in for historical purposes.

### 3.4.4.3. Vector File

A *very* simple vector overlay capability has been added to version 1.15 to support making maps with boundaries on them. The current incarnation paints the overlay data directly into the texture map; it does not overlay the data onscreen as the hex grid and rectangular grid overlays do. Also, this little routine only draws black lines and only processes input from a data file.

I wrote and used this feature to prepare a large-scale map of the local terrain with airspace boundaries on top for work. I calculated the terrain, saved it as a 24-bit image, made a white texture map (sea mask with altitude equal to 100000), then painted the overlay file into the texture map. After saving the mask, I used Photoshop to do the masking and merging to get the airspace boundaries to glow and all of the areas out of the airspace to be dark. After being printed out as a six foot by six foot map, it managed to impress lots of customers.

The input file format is a very simple, one line per record format. All coordinates are in the lat/lon coordinate system, not the pixel coordinate system like many of the other drawing items.

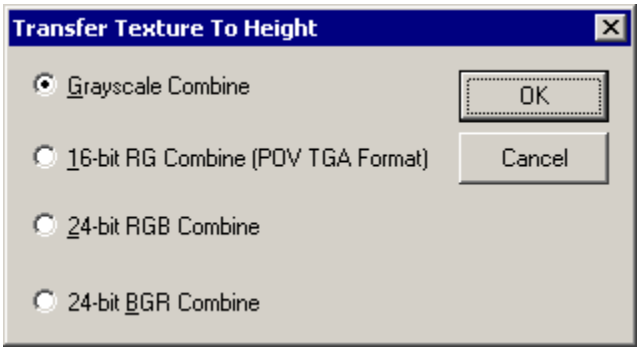
Each record contains one command, such as move, draw, change pen width, or color. Note that the color command doesn't do anything yet. Each record is *very* fixed format: a one-character command, a space (not tab or other item) and a list of up to three numbers, each separated by exactly one space. If this format is not met, the command will likely not be executed. Note that the command code is not case sensitive.

The supported commands are:

| Cmd | Parameters | Example | Description |
|---|---|---|---|
| M | Lat Lon [Alt] | `M -117 35` | Move pen to lat, lon; alt is ignored if present. |
| L | Lat Lon [Alt] | `L -117 34.5` | Draw line using current pen from last line or move position to lat, lon; alt is ignored if present. |
| W | Width | `W 1` | Change the pen width to w pixels. |
| C | Red Green Blue | `C 0 0 0` | Change the pen color to RGB value given. If R, G, B values are all less than or equal to 1.0, then they are scaled by 255; if any are above 1.0, then all are left as specified. Allows both "device-independent" and "device-dependent" operation. |

## 3.4.5. Transfer Texture to Height

Copies the contexts of the texture image to the height field. There are many possible ways to transfer an RGB image into a single height channel, but Wilbur implements just a very few of them.




The RGB channels are combined to create the final height field in the following ways:

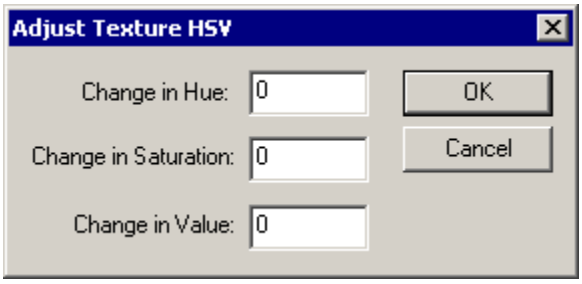
| | |
|---|---|
| Grayscale Combine | 0.3*R + 0.59*G + 0.11f*B |
| 16-bit RG combine | G*256 + R |
| 24-bit RGB Combine | R + 256*G + 256*256*B |
| 24-bit BGR Combine | B + 256*G + 256*256*B |

## 3.4.6. Color Adjustment

Sometimes you want to tweak the colors on the texture without affecting the underlying height data. This menu offers you ways to fill a few of those wants.

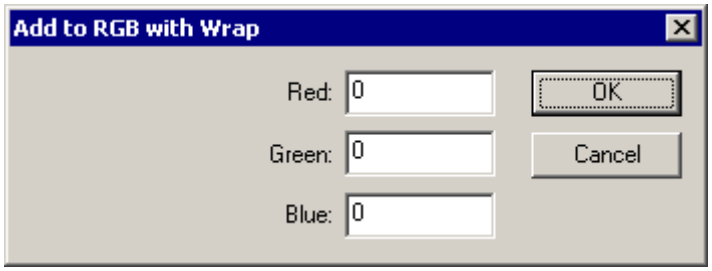
### 3.4.6.1. Adjust HSV

The individual Hue, Saturation, and Value channels of the texture image can be adjusted with this dialog. The operation involves an RGB->HSV transform, the addition of the values indicated in the dialog, and then an HSV->RGB transform.



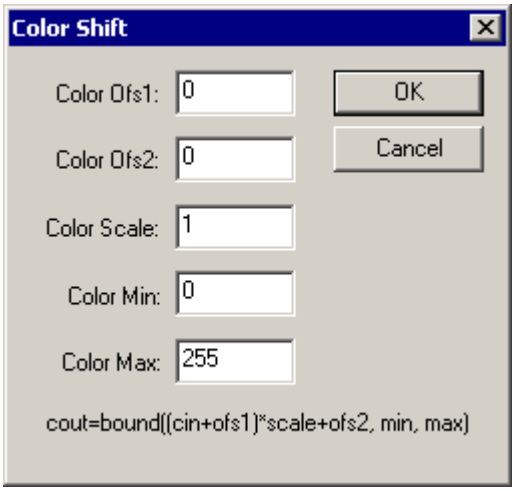

### 3.4.6.2. Add to RGB with Wrap

This operation adds the specified values to each RGB channel in the texture image. If the value exceeds 255, it is wrapped back around starting at 0.



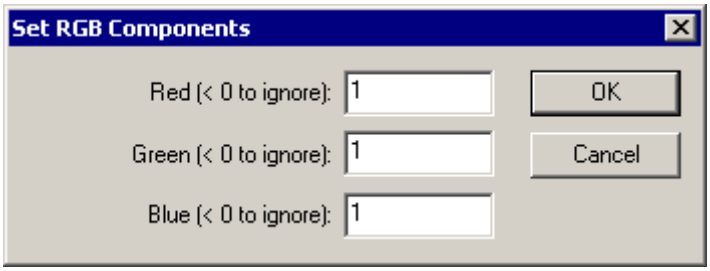

### 3.4.6.3. Color Shift

The equation at the bottom of the dialog pretty much describes the whole function of the operation.



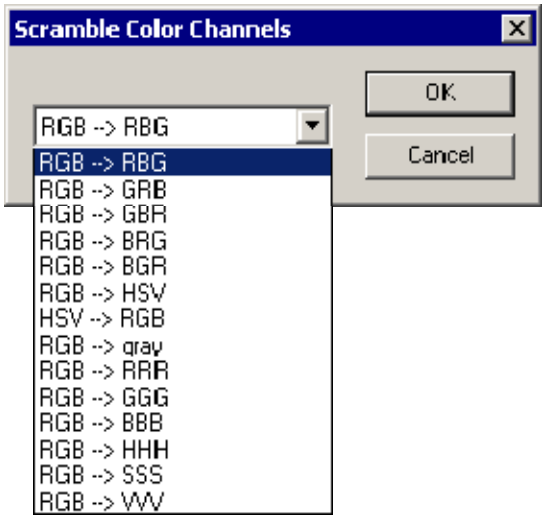

### 3.4.6.4. Set RGB Components

The individual RGB components of all points in the texture can be forced to a particular value using this command. Specifying the full RGB value will clear the whole texture to a particular color. Using a value less than 0 for any of the components will not set that channel. As an example, setting R=-1, G=-1, B=0 will remove all blue contribution from the current texture.



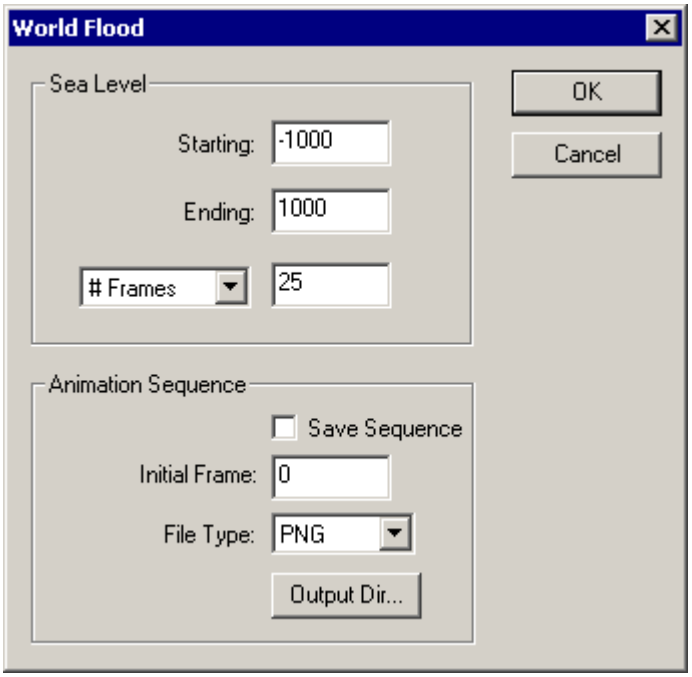

### 3.4.6.5. Transfer Components

This dialog will allow you to transfer color channels around and/or to perform color space conversions. I think the drop list is fairly self-explanatory, but I'm certain somebody will tell me if it's not.



### 3.4.7. World Flood

Remember the really bad Kevin Costner movie Waterworld? Remember the nifty title graphics where the world flooded? Here's your chance to do it yourself. This routine generates a sequence of frames, interpolating sea level either by a set height value or between two endpoints for a given number of frames.




**Starting** and **Ending** are the initial and final sea levels.

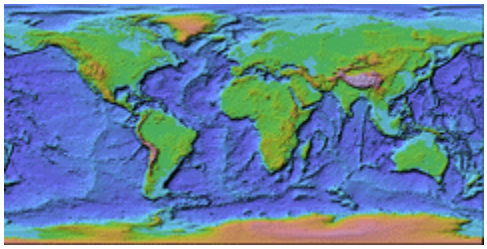
**# Frames** indicates that that number of frames will interpolate between the low and high values. It has the option of being **Sea Delta** in which case sea level starts at starting, and increases by **Sea Delta** amount until it is at **Ending**.

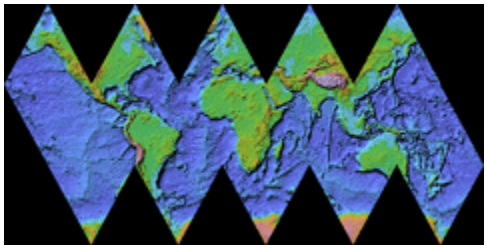
***Animation Sequence*** provides a way to save the images to disk for creating animations later. If **Save Sequence** is checked, the images will be saved. **Initial Frame** is the frame number to start with, and **File Type** is the type of file to generate (currently BMP and PNG only). The files are named PIC00000.*type* where the 00000 is replaced by 00001, etc in each successive frame.

### 3.4.8. Icosahedral Projection

Converts the texture map in memory into an icosahedron (20 sided solid) that can be folded up into something resembling a globe. It does not take into account the proportions of the map or the map edges, just assumes that the displayed area represents a sphere. Because this is just a bitmap-based hack, best effects are obtained if the image is twice as wide as it is high (which means that the triangles for the result will be very close to equilateral).

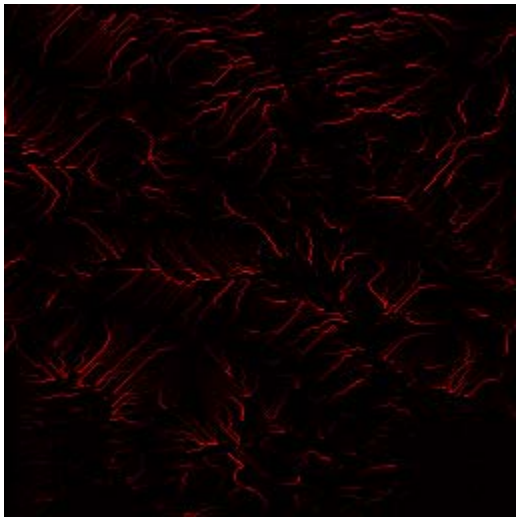
The following pictures show an image of the world before and after the icosahedral projection transform:





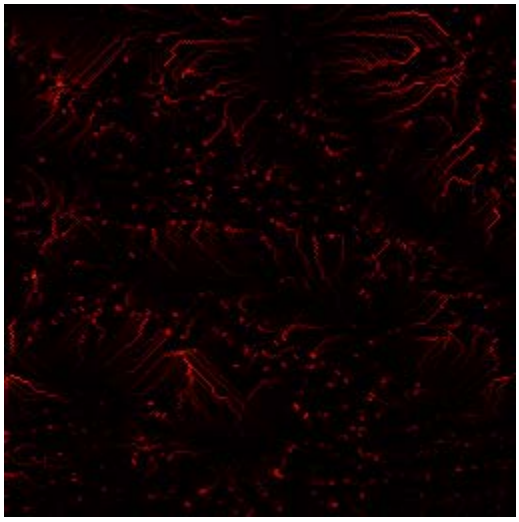
### 3.4.9. Drainage Areas

This operation computes the amount of flow that would occur if water were to flow downhill across the surface. This flow computation does not fill basins or do anything except flow downhill. See 3.4.3.6 for the source image used to compute the drainage image shown below. The amount of flow is computed as an RGG image (see 3.4.5 for a way to transfer this image into a single-valued height field).



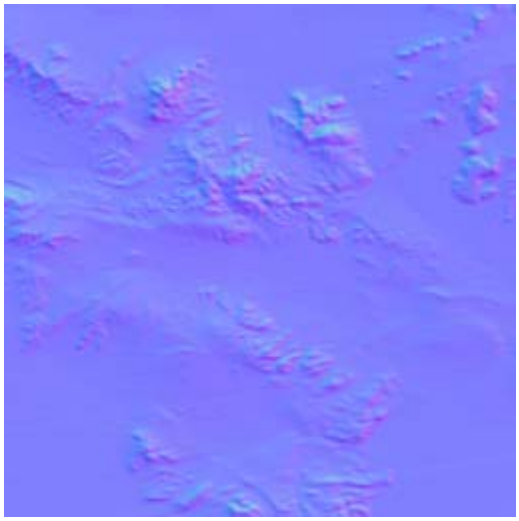
### 3.4.10. Reverse Drainage Areas

This operation is very similar to the operation described in Drainage Areas, above, except that the flow is computed going uphill. Again, the source image is shown in 3.4.3.6.



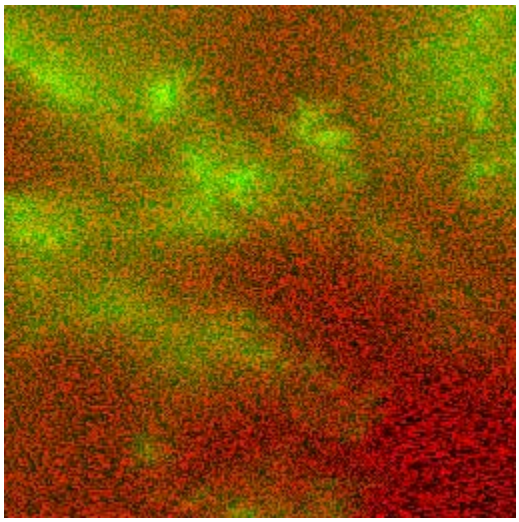
### 3.4.11. Normal Map

In modern 3D graphics contexts, a normal map is an RGB image that specifies the direction of the surface normal vector (a vector representing the direction perpendicular to the base plane of the height field). The normal map for the source image in 3.4.3.6 is shown below. The image below is computed with a vertical scale of 2.



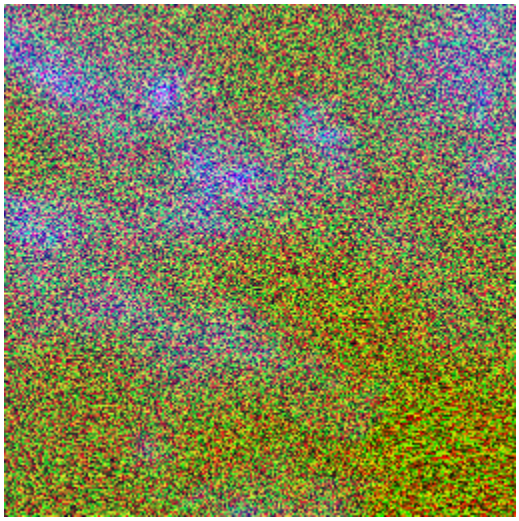
### 3.4.12. 16-bit POV RG Height

This operation is a good deal like 3.4.3.4, except that it spreads the height data across 16 bits in the Red and Green channels rather than just 8 bits in the RGB channels. Long ago the Point-of-View raytracer program stored its height fields in this format (saved in the TGA format). The height field in 3.4.3.6 appears as shown below when rendered as a 16-bit RG height texture.



### 3.4.13. 24-bit RGB

This operation is a good deal like 3.4.3.4, except that it spreads the height data across 24 bits in the Red, Green, and blue channels rather than just 8 bits in the RGB channels. Long ago the Point-of-View raytracer program stored its height fields in this format (saved in the TGA format). The height field in 3.4.3.6 appears as shown below when rendered as a 24-bit RGB height texture.



(Not very pretty, eh?)

## *3.5. Select*

A selection is a mask used to indicate which parts of an image will be used in computations. Wilbur has a relatively rich set of operations for working with selections. Due to an implementation choice, however, selection operations cannot be undone. The selection appears on the surface as an animated dashed outline drawn at the 50% selection level.

### 3.5.1. All

Selects all points in the surface.

### 3.5.2. Deselect

Hides the selection so that all points will be processed in the surface.

### 3.5.3. Reselect

If a previously-active selection was hidden with Deselect, Reselect can be used to recover that selection.

### 3.5.4. Inverse

Changes all unselected points to selected, and all selected points to unselected.

### 3.5.5. Selection Tools

The selection tools allow you to create a new selection or modify an existing one.

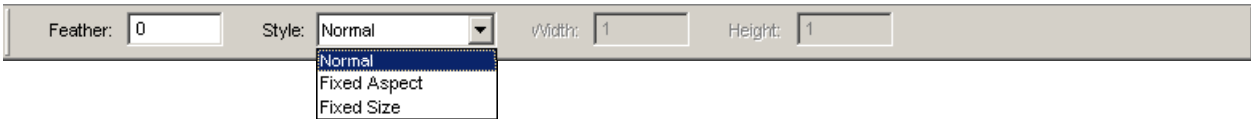
#### 3.5.5.1. Selection Modifiers

For the selection painting tools such as rectangular, elliptical , freehand, and so on, holding down the Shift key when ending the selection will add the selected area to any existing selection. Holding down the Ctrl key when ending the selection will remove the selected area from any existing selection. Holding down no keys will replace any existing selection with the selected area. Holding down both keys will result in an intersection between the existing selection and the new selected area.

In addition to the keyboard modifiers, most selection painting tools provide a toolbar option to feather the selection. You can save a step this way.

### 3.5.5.2. Rectangle Selection Tool

Clicking and dragging will select a rectangular area. The toolbar may be used to force the selection to have a particular width:height ratio (such as a 1:1 for a square) or to force the selection to a particular size. Modifier keys as described in the general selection tools discussion apply.




### 3.5.5.3. Ellipse Selection Tool

Clicking and dragging will select an elliptical area. The toolbar may be used to force the selection to have a particular width:height ratio (such as 1:1 for a circle) or to force the selection to a particular size. Modifier keys as described in the general selection tools discussion apply.

### 3.5.5.4. Freehand Selection Tool

Clicking and dragging will outline an area following the mouse. The area will be selected on mouse button release. The Escape key will abort the selection. Click-release, move, and then clicking again will also work if you'd rather not hold down the mouse button while moving.

### 3.5.5.5. Magic Wand Selection Tool

The Magic Wand selection tool selects an area of similar altitude to that at the initial point clicked. The toolbar is used to control this tolerance. Selected values will be between *alt – tolerance* and *alt + tolerance* where alt is the altitude of the point under the mouse cursor and tolerance is the value specified on the toolbar.


### 3.5.5.6. Move Selection Tool

The Move Selection tool moves the existing selection but will not modify it except for those parts pushed off the edge of the map area. Click and drag to move the selection with the mouse. Use the arrow keys to move the selection one pixel in the direction indicated. Ctrl+arrow will move the selection 10 points in the direction indicated.

### 3.5.5.7. Polygon Selection Tool

This tool is very similar to the freehand selection tool except that you must click for each point in the selection. Right-click to end the selection. The keyboard modifiers will apply to polygon selections.

### 3.5.5.8. Single Row Selection Tool

Selects a single row on the surface. The keyboard modifiers will apply. The intersection modifier is a little odd, however.

### 3.5.5.9. Single Column Selection Tool.

Selects a single column on the surface. The keyboard modifiers will apply. The intersection modifier is a little odd, however.

## 3.5.6. Feather

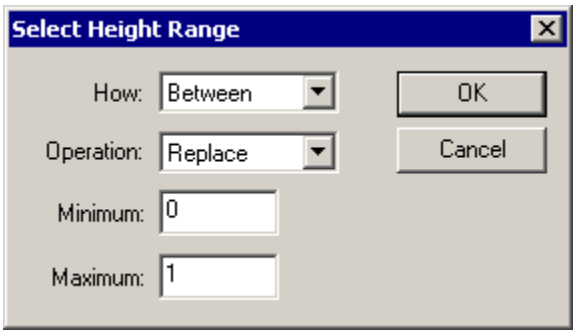
The feather operation smooths out the edges of the selection. This operation asks for a value that represents the area of smoothing. For best effect, apply a blur with half the amount you want, but do it twice. The result will be much smoother than a single, larger blur (Wilbur has a pretty bad blur).

## 3.5.7. From Terrain

Wilbur can create a selection based on the existing data stored in the height field. It can do this either by the altitude value at each point, or by the angle(s) formed by adjacent points.

### 3.5.7.1. Height Range

Allows values in a particular height range to be selected for processing.

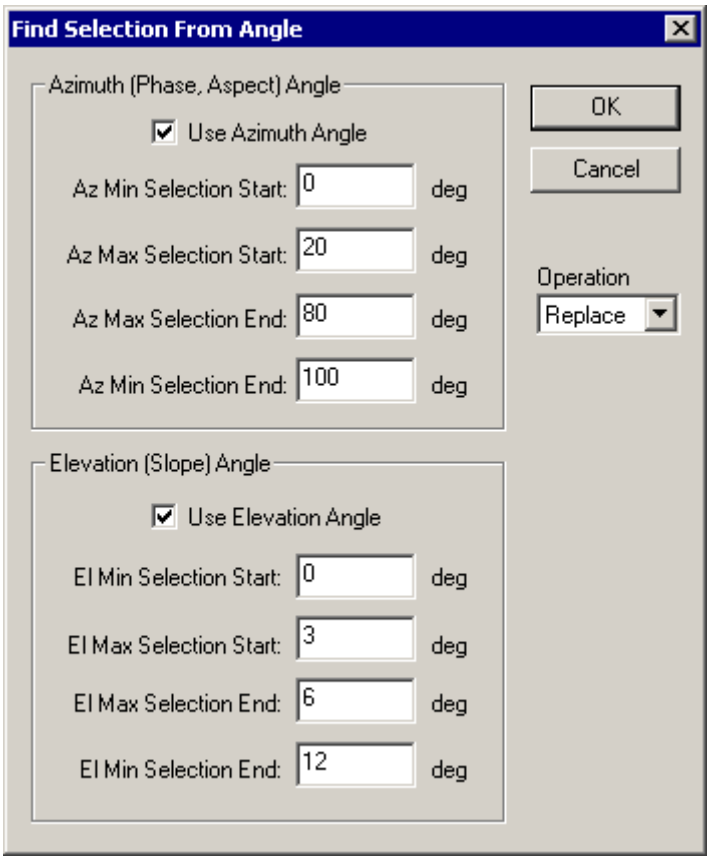



How indicates what Minimum and Maximum mean during the operation.

Operation indicates what will happen with the selected data. Replace replaces the entire selection, while Add adds to the existing selection and Subtract removes from the existing selection.

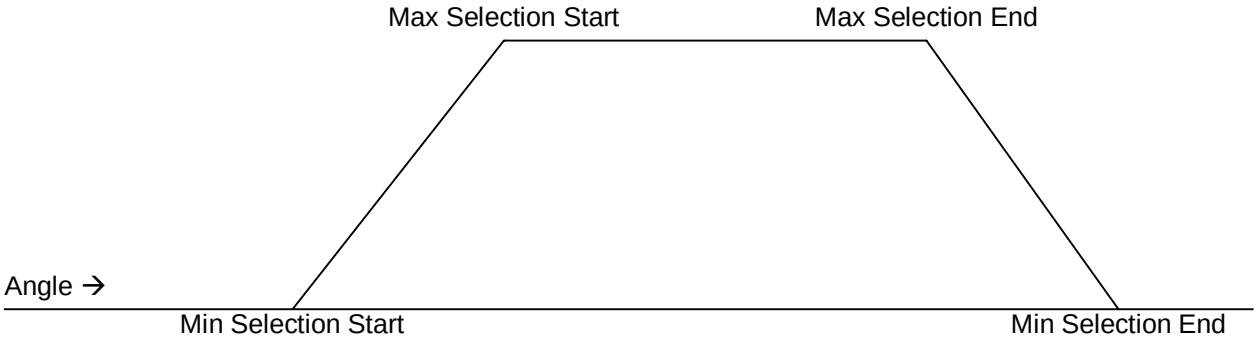
### 3.5.7.2. From Terrain Direction

Allows you to select the terrain based on the direction in which the pixel faces relative to its neighbors.



The *Use* checkboxes indicate if that part will be used. Azimuth indicates the horizontal facing direction, while elevation indicates the vertical facing direction.

*Min Selection Start* is the lowest angle at which the selected area will be selected. *Max Selection Start* is the angle at which the selection starts evaluating at a maximum value. *Max Selection End* is the angle at which the selection stops evaluating at a maximum. *Min Selection End* indicates the ending angle. A picture will probably make this clearer:




*Operation* indicates what will happen with the selected data. Replace replaces the entire selection, while Add adds to the existing selection and Subtract removes from the existing selection.

## 3.5.8. Modify

Some basic operations can be performed on an existing selection.

### 3.5.8.1. Binarize

Many operations convert a hard-edged selection to a soft-edged selection (that's all that Feather does, for example). Binarize will allow you to convert a soft-edged selection back to a hard-edged one. The operation requests a value that will be used as the threshold for the operation. The default displayed threshold for selections is 128, so using that value will get you a hard-edged selection that exactly matches what's shown onscreen.

### 3.5.8.2. Border

Changes the selection to represent a border around the existing selection. The delivered result will usually be matched fairly close to that requested, but may be off by half a pixel one way or another depending on the size of the requested border.

### 3.5.8.3. Expand

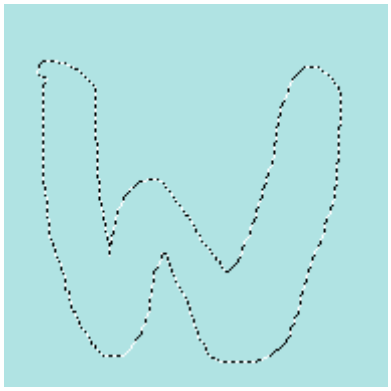
Expands the selection outward by the requested number of pixels. This operation is a morphological dilate operation for those of you that care. Or the max operator. Something like that.

### 3.5.8.4. Contract

Contracts the selection inward by the requested number of pixels. This operation is a morphological erode operation for those of you that care. Or the min operator. Something like that.

### 3.5.8.5. Distance

Computes the distance from the edge of the selection inward. The edge of the selection is likely to look a little odd (it will be too high), so I would recommend applying a feather operation before doing other operations like filling. The image sequence below shows a drawn selection (left) and its appearance after applying the Distance operator. The image on the right shows the effect of using the Set Value operation on the distance selection.



Raw Selection

Distance Operation

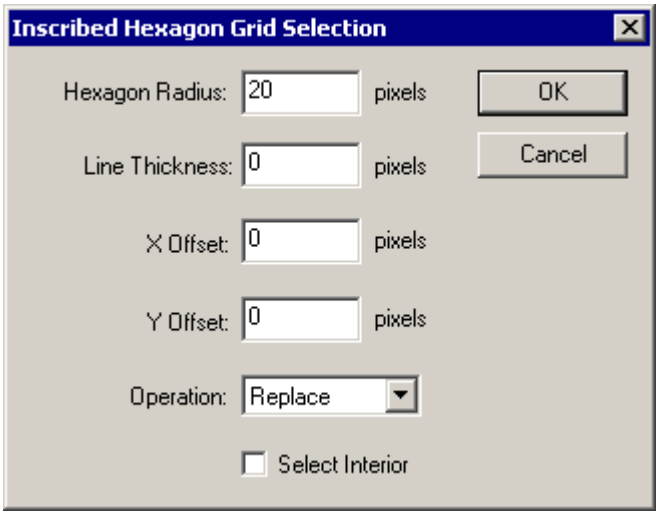
Filled from Distance Selection

### 3.5.9. Render

Some new operations were added to allow rendering a grid into the selection. This grid can be used as the basis of surface adjustment for certain kinds of effects.

#### 3.5.9.1. Hexagon Grid

The hexagon grid draws a hex grid using parameters specified in the dialog shown below:




**Hexagon Radius** is the distance from the center of the hexagon to the corners of the individual hexes.

**Line Thickness** is the thickness of the lines that will be drawn along the edges of the hexagons.

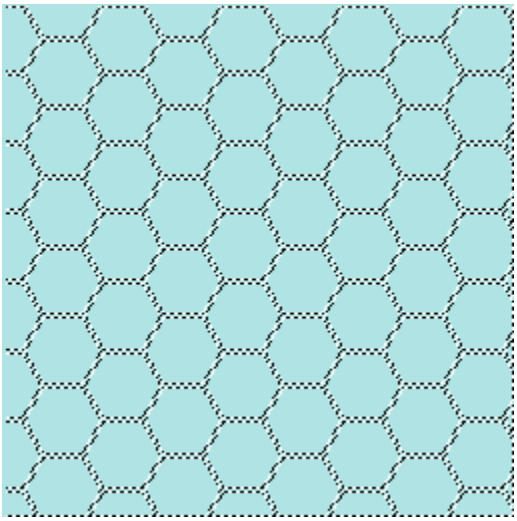
**X Offset** will move the hex grid horizontally by the indicated number of pixels.

**Y Offset** will move the hex grid vertically by the indicated number of pixels.

**Operation** indicates how the hex grid should interact with the existing selection.

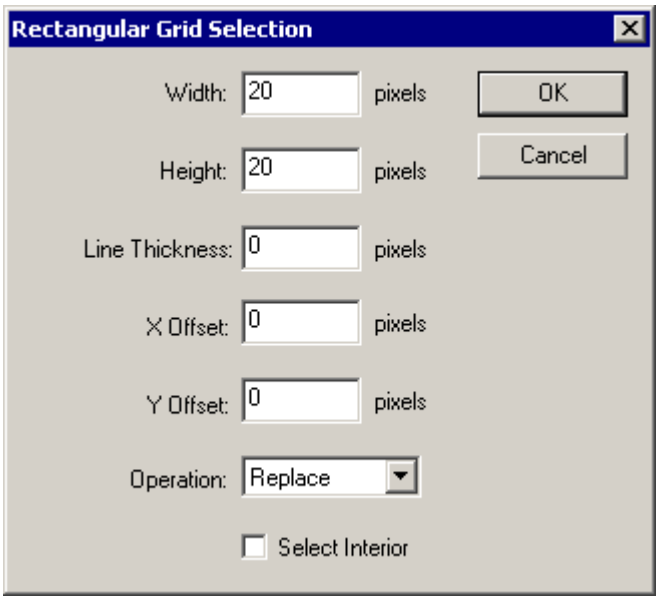
**Select Interior** will select the interiors of the hexes rather than the lines along the hex edges.

The image below shows the effect of a 256x256 surface with the grid values shown in the dialog above.



### 3.5.9.2. Rectangular Grid

The hexagon grid draws a rectangular grid using parameters specified in the dialog shown below:




**Width** is the width of individual rectangles.

**Height** is the height of individual rectangles.

**Line Thickness** is the thickness of the lines that will be drawn along the edges of the rectangles.

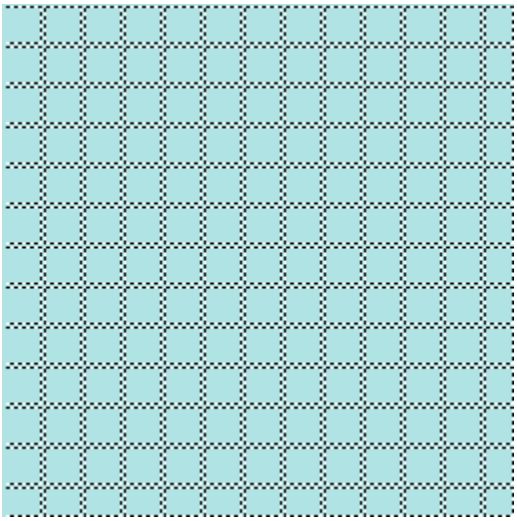
**X Offset** will move the grid horizontally by the indicated number of pixels.

**Y Offset** will move the grid vertically by the indicated number of pixels.

**Operation** indicates how the grid should interact with the existing selection.

**Select Interior** will select the interiors of the rectangles rather than the lines along the hex edges.

The image below shows the effect of a 256x256 surface with the grid values shown in the dialog above.



### 3.5.10. Load Selection

Loads an image from disk and uses it as the selection. The image will be interpreted as a grayscale image and will be stretched to fit the surface. For best results, use a grayscale image the same size as the current image.

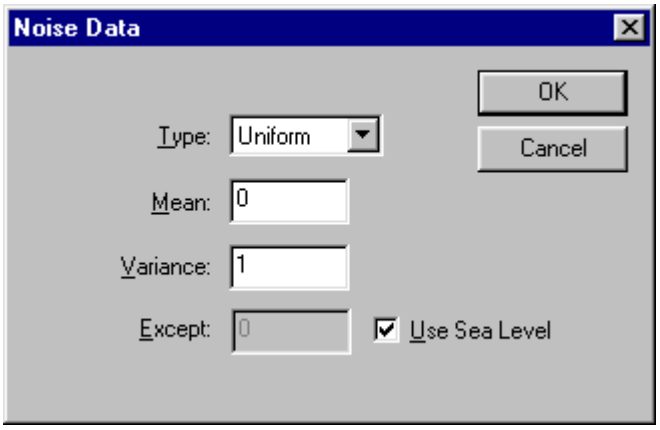
### 3.5.11. Save Selection

Saves the current selection to disk as a BMP file. This selection can be loaded again at a later point.

## *3.6. Filter*

### 3.6.1. Add Noise

Adds noise to the image. The magnitude of the noise can be controlled as can the distribution. The types of noise available are Uniform and Gaussian. Any other types listed in the dialog don't work at the moment.




**Type** is the kind of noise to be added to the surface. Available types are *Uniform*, *Gaussian*, *Rayleigh*, and *Exponential*. Only *Uniform* and *Gaussian* work properly at the moment, though.

**Mean** is the average value for the noise. A value of 0 will cause the noise to be centered around 0 (both positive and negative values), for example.

**Variance** is the magnitude of the noise from the center point.

If **Use Sea Level** is checked, the noise will be added to all points except the currently defined sea level. If not checked, **Except** will be available.

**Except** indicates the level to which noise will not be added. This feature allows, for example, noise to each point except the sea (to help get rid of some peculiar flat areas from certain data sets).

### 3.6.2. Blur

#### 3.6.2.1. Gaussian Blur

Performs a Blur of arbitrary size. The entered value is not a pixel radius, but rather about 2.5 times less than the desired radius. Effective values range from 0.5 (the minimum effect) to large values (using a value more than twice the size of the image is not particularly useful).

#### 3.6.2.2. Some (3x3)

Uses the general convolution operation with a 3x3 blur kernel.

#### 3.6.2.3. More (5x5)

Uses the general convolution operation with a 5x5 blur kernel.

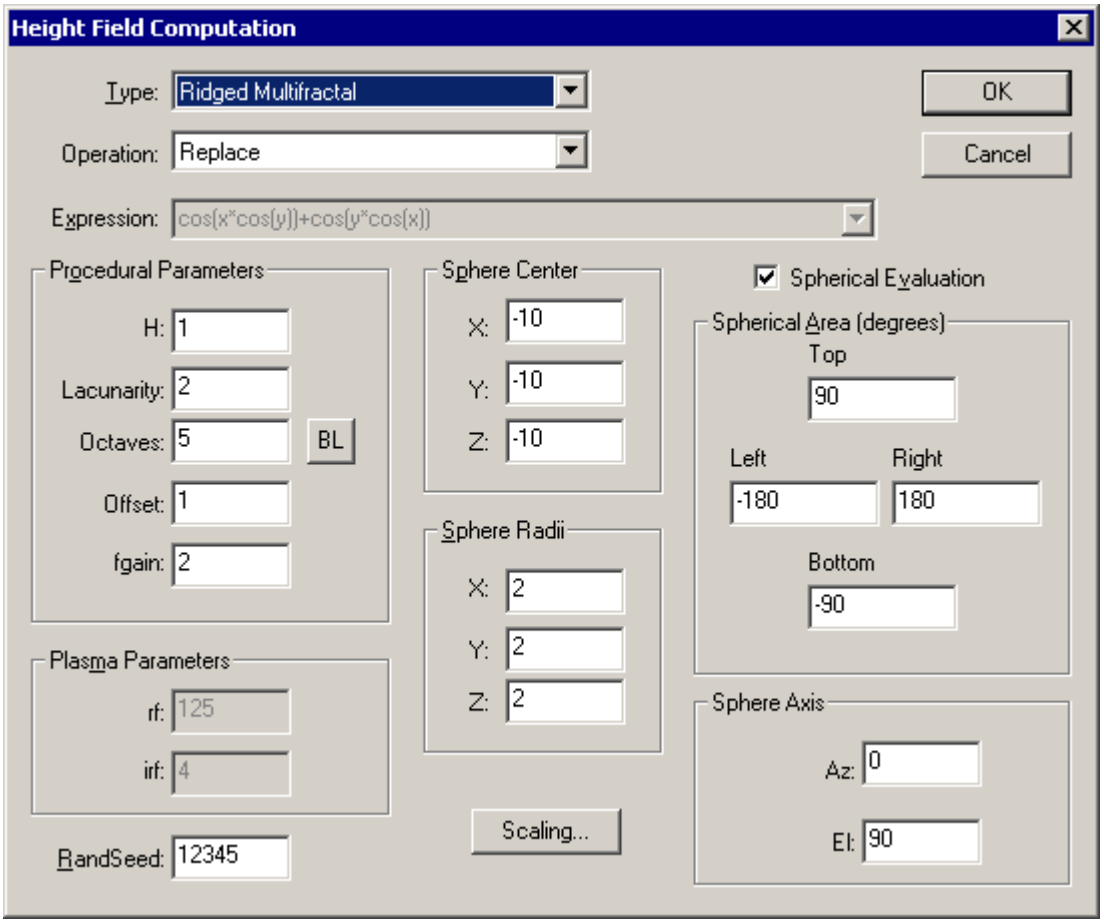
#### 3.6.2.4. Lots (7x7)

Uses the general convolution operation with a 7x7 blur kernel.

### 3.6.3. Calculate Height Field

Wilbur can generate purely synthetic height fields using several algorithms. Some of the algorithms are from Ken Musgrave's sections in the book "Texturing an Modeling, a Procedural Approach" published by Academic press (ISBN 0-12-228760-6). It's a good book; consider getting a copy.

The menu item calls the dialog below. Not all of the items are used for all of the modes. Not the best way to handle the situation, but it's easier than doing it right.




**Type** shows the basic surface type that will be calculated. There are currently 7 supported types of surfaces: Plasma, Math Function, Fractional Brownian Motion, Multifractal, Hetero Terrain, Hybrid Multifractal, and Ridged Multifractal. Each of these types is discussed in detail below, along with what data each needs to work its magic.

**Operation** indicates how the new height field will be combined with the existing one. The default operation is to replace the existing height field.

**Expression** shows the current expression used to evaluate the Math Function surface type. The list can be dropped down to show other recent expression used. Expressions are stored in the registry and are never discarded. Editing the list can get to be a pain, so I usually just delete the "Wilbur" registry key once in a while. Fiddling with the registry can be bad for inexperienced users, so be careful. I assume no responsibility if you manage to make you machine not work right after *you* messed with the registry (see the "Legal Bits" section up top).

**H** deals with Musgrave's procedural terrains and "determines the fractal dimension of the roughest areas". See Chapter 9 of the book for a complete description.

**Lacunarity** deals with Musgrave's procedural terrain and "is the gap between successive frequencies". See Chapter 9 of the book for a complete description.

**Octaves** deals with Musgrave's procedural terrain and "is the number of frequencies" in the surface. This is the iteration count, which may be non-integral but must be positive. The more octaves used, the longer the surface will take to calculate. Using twice as many octaves takes four times as long to calculate. See Chapter 9 of the book for a complete description.

**Offset** deals with Musgrave's procedural terrain and "raises the terrain from 'sea level'". It is an offset to the 3D noise function used as the basis function for the terrains to move the value from the [-1, +1] range. .

**fgain** deals with Musgrave's procedural terrain. It isn't described too well and applies only to the "Ridged Multifractal" type. Changing its value can result in truly spectacular results, however.

**RandSeed** is the seed for the random number generator used for all of the surface types. Entering the same value here will cause the same sequence of values to be used, causing the exact same surface to be formed. Changing the value will result in different surfaces.

**rf** and **irf** deal with plasma surfaces only. They control the statistics of the surface.

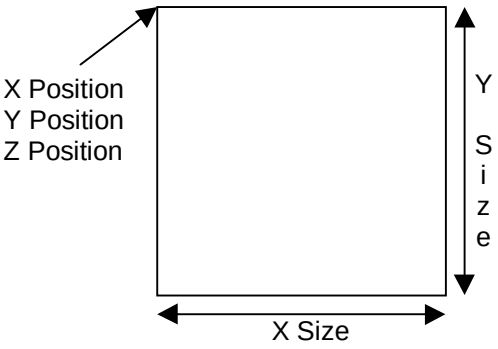
The **Scaling** button allows the surface to be scaled IAW the user's wishes as part of the surface calculation. Clicking the button brings up the scaling dialog as described in 3.6.7.3.

**Position** is the starting location for the surfaces. X, Y and Z are used in all of the Musgrave terrain to control where the 3D noise function is evaluated. The Z parameter isn't used in other terrain types. When the "Spherical Evaluation" checkbox is checked, these values are the center point of the ellipsoid used in the calculations.

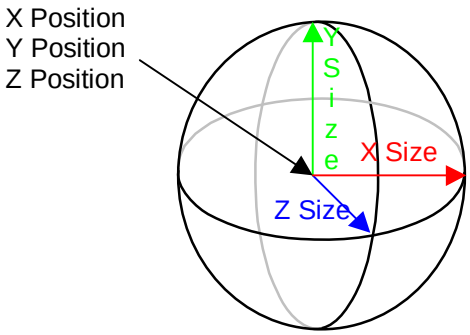
**Size** gives the size of the surface (width = X, height = Y, depth = Z). The surface is evaluated from (X Position, Y Position) to (X Position + X Size, Y Position + Y Size). The Z parameter isn't used by any of the surface types yet. When the "Spherical Evaluation" checkbox is checked, these values are the radii of the ellipsoid used in the calculations.

The ***Spherical Area*** parameters are the portions of a sphere used for calculations. When "Spherical Evaluation" is checked, the software evaluates the function over a sphere, not a planar section. Because the Position and Size values go into describing the spheroid, a new set of parameters is needed to describe the area of evaluation on the sphere's surface. All values are in degrees.

The **Spherical Evaluation** checkbox determines how the system evaluates its functions. How the parameters are used is shown below.




Without "Spherical Evaluation "

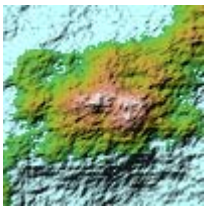



With "Spherical Evaluation"

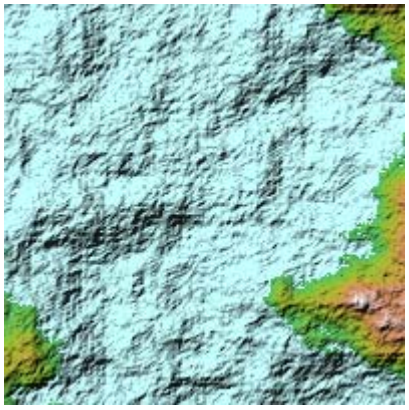
The ***Sphere Axis*** group indicates the point on the spherical surface through which the north pole should point.

### 3.6.3.1. Plasma Fractal

A variation on the classic plasma algorithm, this one interpolates along the edges, then subdivides along each line. The surface has two undesirable properties, though: it isn't easy to extend past the current surface (even chaining the resolution will change the appearance) and it has awful creases along the X and Y directions. The pictures below show some examples of this type of surface:



100x100, Seed=12345, rf=125, irf=4



200x200, all others same as at left

The entries which affect this surface are RandSeed, rf, and irf. No other options are used in calculating this type of surface. RandSeed controls the sequence of random numbers used to generate the surface. The rf and irf parameters control the statistics of the terrain. Play with them, enjoy them, make them your friend. Together, they can help to defeat creasing.

### 3.6.3.2. Math Function

Arbitrary math functions of the form z = f(x,y) can be entered. The function f(x,y) is parsed within the program and evaluated at run-time. The current version allows for expressions that contain functions and operators. Functions take either 0 parameters (e.g. x, y, r, pi) or 1 parameter (e.g. sin(a), cos(a), atan(a)). Functions with 0 parameters are referred to as "variables" below.

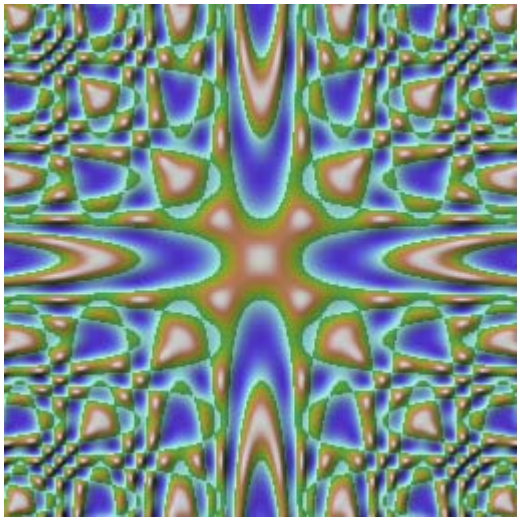
The available operators are add (+), subtract (-), multiply (*), divide (/), remainder (%), and exponent(^). Predefined functions are:

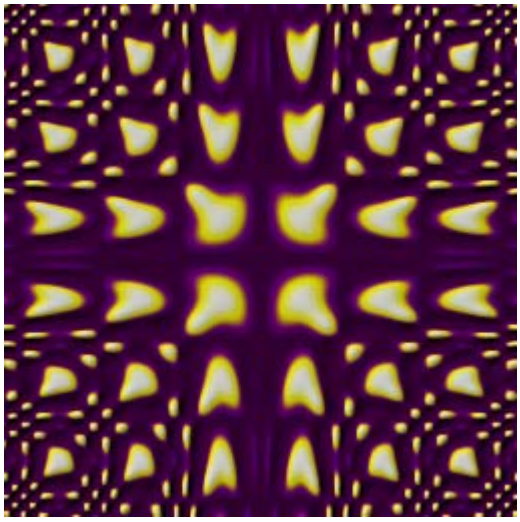
| Function Name | Syntax | Variable Description |
|---|---|---|
| Abs | Abs(a) | Absolute value of the argument |
| Acos | Acos(a) | Arccosine of the argument. The argument range is -1 to 1. |
| And | And(a,b) | Returns 1 if both a and b are non-zero, 0 otherwise. |
| Asin | Asin(a) | Arcsine of the argument. The argument range is -1 to 1. |
| Atan | Atan(a) | Arctangent of the argument. |
| Atan2 | Atan2(y,x) | Returns the arctangent of y / x, adjusted for quadrant. |
| Besselj | Besselj(n,x) | Bessel function of the first kind of order n. (if it's in the VC++ math library, you get it, no matter how strange) |
| Bessely | Bessely(n,x) | Bessel function of the second kind of order n. (see above) |
| Bias | Bias(b,x) | Returns x^(ln(b)/ln(0.5)). |
| Ceil | Ceil(a) | Smallest integer that is greater than or equal to the argument. |
| Cos | Cos(a) | Cosine of the argument. The argument is in radians. |
| Cosh | Cosh(a) | Hyperbolic cosine of the argument |
| Crater | Crater(r, RimDist, RimHeight, BowlBase, BowlSteep) | Calculates a crater-like function. R is the argument for the function; RimDist is the distance from the crater center to the highest point on the rim; RimHeight is the height above the baseline for the rim; BowlBase is the lowest point of the basin; BowlSteep is an exponential control for wall steepness/basin flatness. This function is defined on the interval 0 to 1 (inclusive). R and RimDist are expressed as a number from 0 to |

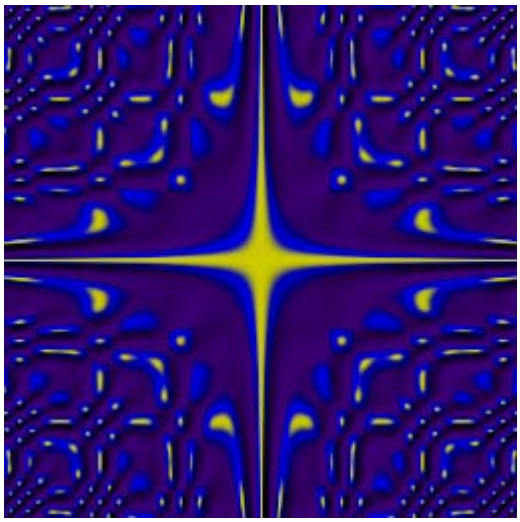
| | | |
|---|---|---|
| | | 1 (R must be mapped back to the range 0-1 during evaluation); RimHeight and BowlBase may have any value deemed appropriate. |
| Curval | Curval | Adds the current value of the surface at this point (useful for accumulating operations). |
| Dist | Dist<br>Dist(z)<br>Dist(x,y)<br>Dist(x,y,z)<br>Dist(x1, y1, x2, y2)<br>Dist(x1, y1, z1, x2, y2, z2) | Returns distance from 0,0 to current point (same as r)<br>Returns distance from 0,0,0 to current point at elevation z.<br>Returns distance from 0,0 to specified XY point<br>Returns distance from 0,0 to specified XYZ point<br>Returns the distance from point x1,y1 to point x2,y2<br>Returns the distance from point x1,y1,z1 to point x2,y2,z2 |
| E | E | Value of the base of natural logarithms (e = 2.718281828459045… to the full numeric precision of the machine). |
| Eq | Eq(a,b) | Returns 1 if a is equal to b, 0 otherwise. |
| Exp | Exp(a) | The value of e raised to the x power. |
| Facing | Facing<br>Facing(ix, iy) | Returns a value from 0-1 that specifies the azimuth (facing) angle of the surface point indicated. A value of 0 is north, and goes clockwise to 1 (0.25 is east, 0.5 is south, 0.75 is west, and 1.0 is north again).<br>ix, iy specify the integer lattice location (the array indices, from 0..map width for ix and 0..map height for iy). The form without any parameters uses the current point. |
| Fbm | Fbm<br>Fbm(x,y,z)<br>Fbm(x,y,z,h)<br>Fbm(x,y,z,h,l)<br>Fbm(x,y,z,h,l,o) | Adds the fractional brownian motion texture to the surface. If no parameters are entered, the current ones entered in the system dialog will be used.<br>x,y,z specify the XYZ point in space at which to evaluate; h is the fractal dimension; l is the lacunarity (roughness); o is the number of octaves (complexity). |
| Floor | Floor(a) | Largest integer less than or equal to the argument. |
| Gain | Gain(g,x) | If (x < 0.5) returns bias(1-g,2*x)/2<br>else returns 1-bias(1-g,2-2*x)/2 |
| Ge | Ge(a,b) | Returns 1 if a is greater than or equal to b, 0 otherwise. |
| Gt | Gt(a,b) | Returns 1 if a is greater than b, 0 otherwise. |
| Hetero | Hetero<br>Hetero(x,y,z)<br>Hetero(x,y,z,h)<br>Hetero(x,y,z,h,l)<br>Hetero(x,y,z,h,l,o)<br>Hetero(x,y,z,h,l,o,f) | Adds the hetero terrain texture to the surface. If no parameters are entered, the current ones entered in the system dialog will be used.<br>x,y,z specify the XYZ point in space at which to evaluate; h is the fractal dimension; l is the lacunarity (roughness); o is the number of octaves (complexity); and f is the offset parameter. |
| Hybrid | Hybrid<br>Hybrid(x,y,z)<br>Hybrid(x,y,z,h)<br>Hybrid(x,y,z,h,l)<br>Hybrid(x,y,z,h,l,o) | Adds the hybrid multifractal texture to the surface. If no parameters are entered, the current ones entered in the system dialog will be used.<br>x,y,z specify the XYZ point in space at which to evaluate; h is the fractal dimension; l is the lacunarity (roughness); o is the number of octaves (complexity); and f is the offset parameter. |

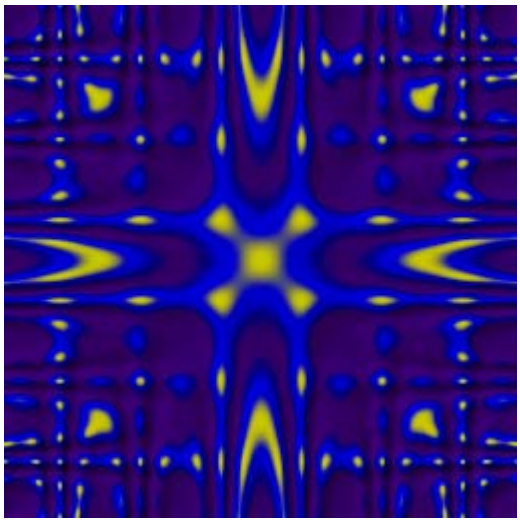
| | | |
|---|---|---|
| | Hybrid(x,y,z,h,l,o,f) | |
| If | If(b,r1,r2) | Returns r1 if b is non-zero, otherwise r2. Very similar to the C construct b?r1:r2. |
| Le | Le(a,b) | Returns 1 if a is less than or equal to b, 0 otherwise. |
| Ln | Ln(a) | Natural logarithm (base e) of the argument. Negative values are converted to positive before calculation. An argument of 0 returns a large negative number. |
| Log | Log(a) | Common logarithm (base 10) of the argument. Negative values are converted to positive before calculation. An argument of 0 returns a large negative number. |
| Lt | Lt(a,b) | Returns 1 if a is less than b, 0 otherwise. |
| Multi | Multi<br>Multi(x,y,z)<br>Multi(x,y,z,h)<br>Multi(x,y,z,h,l)<br>Multi(x,y,z,h,l,o)<br>Multi(x,y,z,h,l,o,f)<br>Multi(x,y,z,h,l,o,f,g) | Adds the multifractal texture to the surface. If no parameters are entered, the current ones entered in the system dialog will be used.<br>x,y,z specify the XYZ point in space at which to evaluate; h is the fractal dimension; l is the lacunarity (roughness); o is the number of octaves (complexity); f is the offset parameter, and g is the fgain parameter. |
| Noise | Noise(x)<br>Noise(x,y)<br>Noise(x,y,z)<br>Noise(x,y,z,t) | Calculates a noise value in the number of dimensions given (1, 2, 3, or 4). |
| Or | Or(a,b) | Returns 1 if either a or b or both are non-zero, 0 otherwise. |
| Pi | Pi | Value of pi (3.14159265359… to the full numeric precision of the machine). |
| Qfract | Qfract<br>Qfract(zr,zi)<br>Qfract(zr,zi,cr,ci)<br>Qfract(zr,zi,cr,ci,rng,iter) | Calculates a fractal of the form z=z*z+c (Mandelbrot or Julia set variants). In the lists at left, zr and zi are the real and imaginary parts of the z value, cr and ci are the real and imaginary parts of the c value, rng is the escape radius (2 is the "normal" value), and iter is the maximum number of iterations. Use Qfract(0,0,x,y) to calculate a Mandelbrot set and Qfract(x,y,0,0) to calculate the Julia set. |
| R | R | Distance from 0,0 to the current x,y location. Equivalent to sqrt(x*x+y*y) |
| Rand | Rand(a) | Uniformly distributed random number between 0 and the argument. |
| Rem | Rem(a) | Non-integer portion of the argument. Equal to fmod(a, 1.0). |
| Ridged | Ridged<br>Ridged(x,y,z)<br>Ridged(x,y,z,h)<br>Ridged(x,y,z,h,l)<br>Ridged(x,y,z,h,l,o)<br>Ridged(x,y,z,h,l,o,f)<br>Ridged(x,y,z,h,l,o,f,g) | Adds the ridged multifractal texture to the surface. If no parameters are entered, the current ones entered in the system dialog will be used.<br>x,y,z specify the XYZ point in space at which to evaluate; h is the fractal dimension; l is the lacunarity (roughness); o is the number of octaves (complexity); f is the offset parameter, and g is the fgain parameter. |

| | | |
|---|---|---|
| scurve | Scurve(x) | Calculates a nice cubic blending curve. Defined from 0 to 1. |
| Slope | Slope<br>Slope(ix, iy) | Returns a value from 0-1 that specifies the elevation (slope) angle of the surface point indicated. A value of 0 is horizontal and a value of 1 is vertical.<br>ix, iy specify the integer lattice location (the array indices, from 0..map width for ix and 0..map height for iy). The form without any parameters uses the current point. |
| Sin | Sin(a) | Sine of the argument. |
| Sinc | Sinc(a) | Returns sin(a)/a except at 0; returns 1 at 0. |
| Sqrt | Sqrt(a) | Square root of the argument. |
| T | T | Angle from 0,0 to the current location. The angle returned is in radians going from -pi to pi. Equivalent to atan(y/x), corrected for quadrant and checking for x = 0. |
| Tan | Tan(a) | Tangent of the argument. |
| Tanh | Tanh(a) | Hyperbolic tangent of the argument. |
| X | X | Current horizontal (longitude) location on the surface. |
| Xmax | Xmax | Maximum x value for evaluation |
| Xmin | Xmin | Minimum x value for evaluation |
| Xor | Xor(a,b) | Returns 1 if either a or b (not both) are non-zero, 0 otherwise. |
| Xrot | Xrot(t)<br>Xrot(t,r)<br>Xrot(t,r,x,y) | Rotates/scales a point around another the origin and returns the x component.<br>Parameters: t is the angle in radians to rotate counter-clockwise; r is the radius scaling factor (1=no scale); x and y are the components of the point to rotate. |
| Xsize | Xsize | Width of the surface (xmax - xmin) |
| Y | Y | Current vertical (latitude) location on the surface. |
| Ymax | Ymax | Maximum y value for evaluation |
| Ymin | Ymin | Minimum y value for evaluation |
| Yrot | Yrot(t)<br>Yrot(t,r)<br>Yrot(t,r,x,y) | Rotates/scales a point around another the origin and returns the y component.<br>Parameters: t is the angle in radians to rotate counter-clockwise; r is the radius scaling factor (1=no scale); x and y are the components of the point to rotate. |
| Ysize | Ysize | Height of the surface (ymax - ymin) |
| Zmax | Zmax | Current maximum point on the surface |
| Zmin | Zmin | Current minimum point on the surface |


cos(x*cos(y))*cos(y*cos(x))


sin(x*cos(y))*sin(y*cos(x))

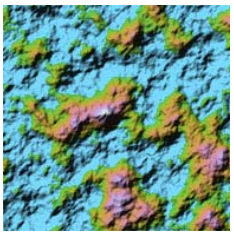

cos(x*sin(y))*cos(y*sin(x))

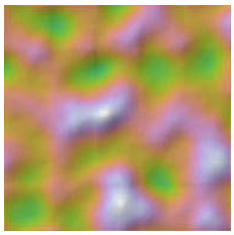

cos(x*cos(y))+cos(y*cos(x))

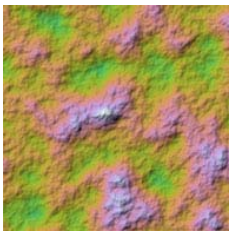
### 3.6.3.3. Textures from the book "Textures and Modeling: A Procedural Approach"

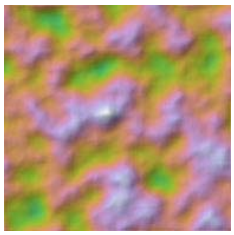
The following types of surface are all taken Ken Musgrave's sections of the book "Textures and Modeling: A Procedural Approach". Immediately below is a quick example of each of these surface types using the parameters shown in the dialog below.

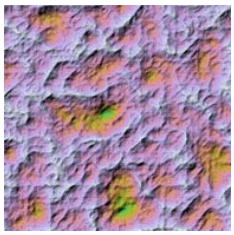
The similarity in overall pattern is a function of the underlying noise function (and random number generator).


fBm


Multifractal


Hetero Terrain


Hybrid Multifractal


Ridged Multifractal

#### 3.6.3.3.1. Fractional Brownian Motion (fBm)

The fBm terrain is similar to the plasma terrain, but has two distinct advantages: it has no creases and it forms a continuous space that can be panned around and changed in scale without affecting the appearance. The parameters used for this surface are H, Lacunarity, and Octaves.

#### 3.6.3.3.2. Multifractal

This texture is very similar to fBm, but uses the offset parameter as well.

#### 3.6.3.3.3. Hetero Terrain

The Hetero terrain is also like fBm, but it also uses the offset parameter.

#### 3.6.3.3.4. Hybrid Multifractal

Yes, it's getting monotonous, but the Hybrid Multifractal terrain is also like fBm, but it also uses the offset parameter.

#### 3.6.3.3.5. Ridged Multifractal

The Ridged Multifractal terrain is different because it uses the fgain parameter as well.

### 3.6.4. Erosion

Wilbur doesn’t correctly implement a true fluvial (river-type) erosion model. For best effects, I would recommend these steps:

Add 10% noise (Filter>>Add Noise)

Fill basins (Filter>>Fill>>Fille Basins, -1 as the parameter).

Add 10% noise (Filter>>Add Noise)

Fill basins (Filter>>Fill>>Fille Basins, -1 as the parameter).

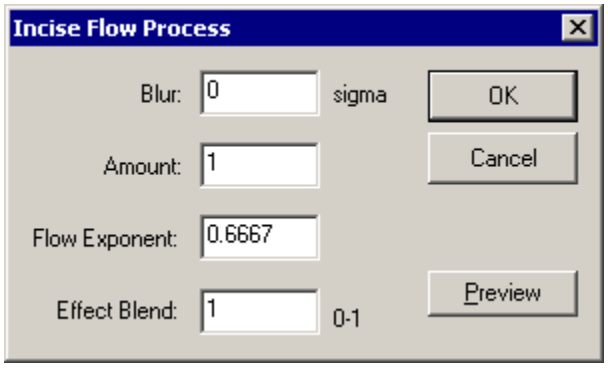
Incise Flow (Filter>>Erosion>>Incise Flow)

#### 3.6.4.1. Erosion Cycle

This operation is a very simple attempt at packaging some of the basic items needed to get something that looks like erosion working. It does a basin fill followed by an incise flow operation. It lets you continue repeating this operation over and over. Not as many controls for the flow as the current Incise Flow operation, but it is a little easier to do some things.

#### 3.6.4.2. Incise Flow

Incise flow calculates a flow pattern across the surface and the removes altitude from the surface by an amount proportional to the flow amount and altitude at that point.

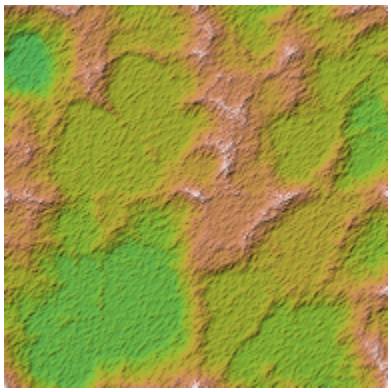



Blur is the amount of blur that will be applied to the flow pattern. Larger blur amounts result in more diffuse river networks.

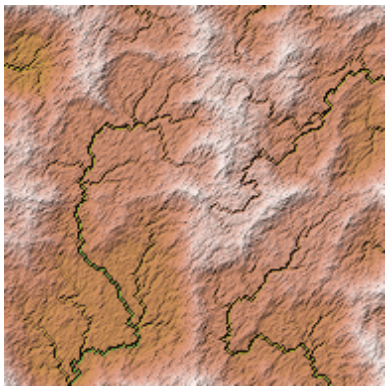
Amount is the amount of material that will be removed. Values smaller than 1 will give progressively less effect for the river channel.

Flow Exponent is the amount by which the flow values will be modified before being used. Smaller values (closer to 0) will give a more extensive river network (more channels visible), while larger values will result in fewer river channels.

Effect Blend is the amount by which the resulting incised flow pattern will be blended with the original surface data. A value of 0.5 will be half from each. Using values less than 0 will result in the flow pattern being raised rather than incised.



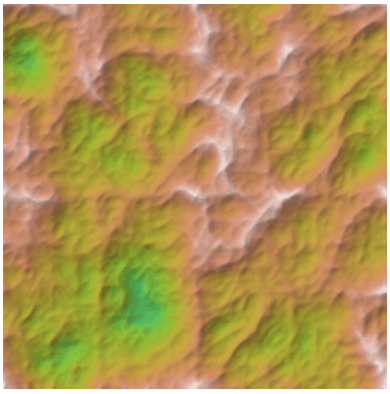
Base Surface (basin fill + 10% noise + basin fill)



Incise with default parameters

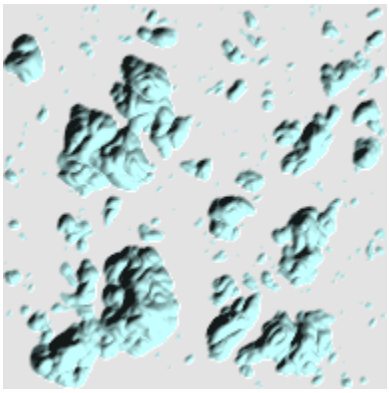
### 3.6.5. Fill

The image shown below is the base surface used for the discussions of the fill types.



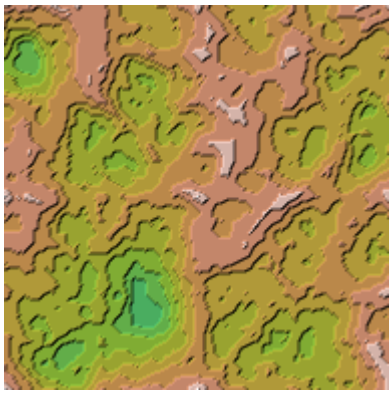
### 3.6.5.1. Compute Basin Deltas

The Fill Basins operation will fill in the basins, but they look a little flat. This operation will compute those same basins, but it returns the depth of material required to fill the basin. An image-editing program such as Photoshop can be used to merge the basin delta mask with the terrain surface to yield a set of lakes. It's a bit tedious but still doable.

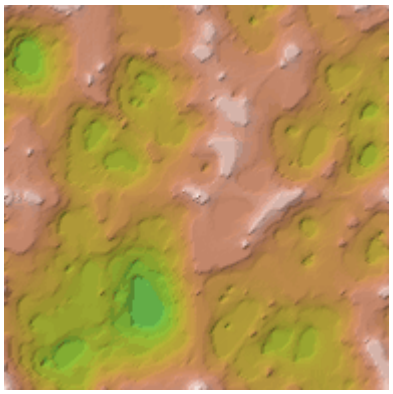


### 3.6.5.2. Deterrace

Deterrace reduces the effects of stairsteps in the image. The implemented algorithm isn't ideal but it does reduce the effects a little bit. The worst effects are seen at the high and low flat areas (basins and crests).



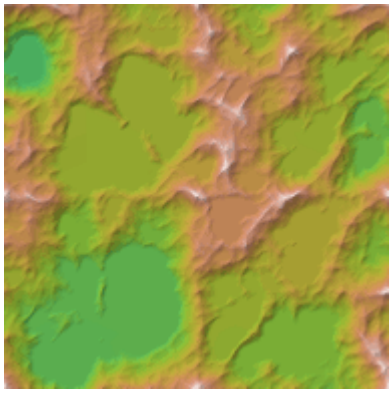
10-level thresholded image



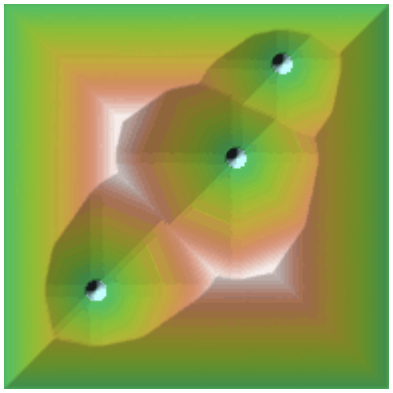
After deterrace

### 3.6.5.3. Fill Basins

Olivier Planchon wrote a good paper on filling basins. I stole the sample code (used with permission so it's not exactly stealing, I suppose) and modified it a little. The result is Wilbur's Fill Basins code. The parameter indicates the amount by which each step in from the edge of the map or sea level increases. A value of 0 provides completely flat results, while a value of 1 provides 45-degree rising slopes. Specifying a value less than 0 (such as –1) calculates a value sufficient to get a definite slope but small enough to appear flat even on large surfaces.

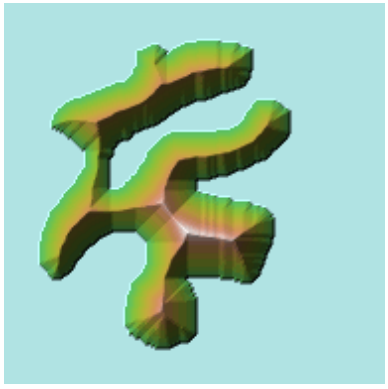


Fill Basins, value = 0



Fill Basins, value = 1 with internal holes below sea level.

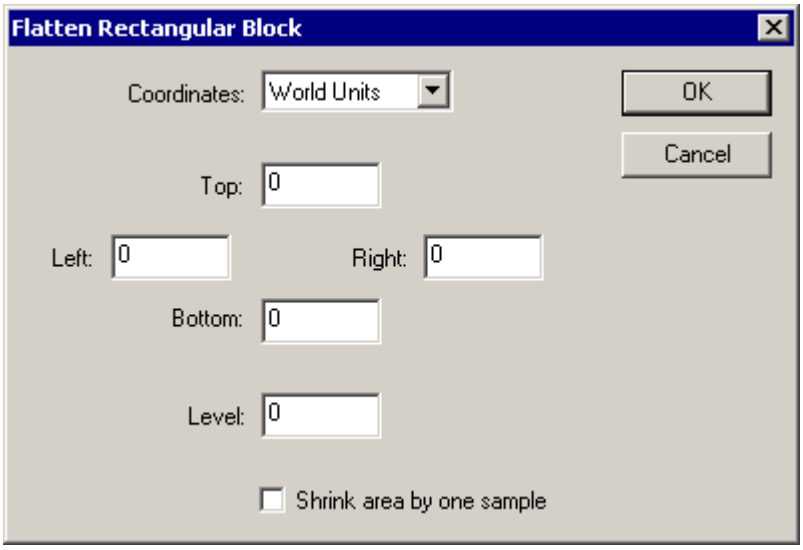
Simple shape above sea level



Fill Basins, value = 1

### 3.6.5.4. Flatten Block

Flatten Block can be used to flatten a rectangular area on the surface.




**Coordinates** is used to specify the units used for Top, Left, Right, and Bottom. World Units treats the values as world units, while Pixels uses the sample index within the surface.

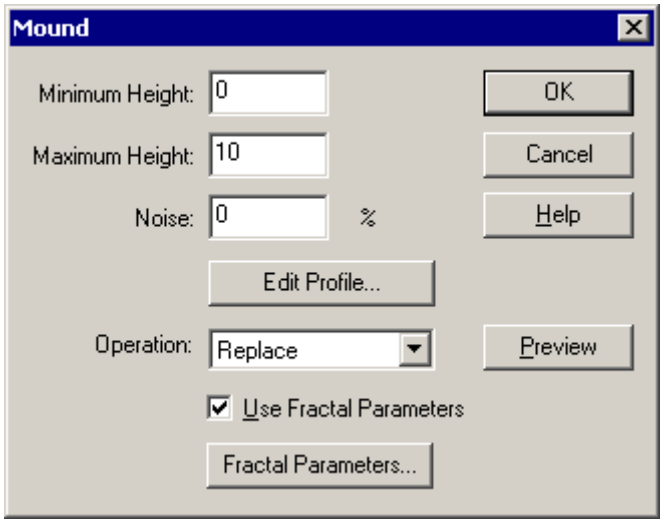
**Top**, **Left**, **Right**, and **Bottom** specify the edges of the block to set to **Level**.

**Level** specifies the altitude at which the block will be set.

**Shrink area by one sample** will reduce the flattened area by one surface point sample in each direction. Shrinking the area was useful for getting better blends when I used this operation to punch out areas to replace with higher-resolution data for a 3D modeling program.

### 3.6.5.5. Mound

The mound operation fills the selection with a blob that reflects the outlines of the current selection.




**Minimum Height** is the height of the lowest point in the mound, in meters.

**Maximum Height** is the height of the highest point in the mound, in meters.

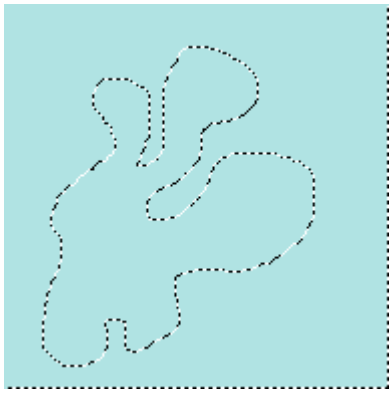
**Noise** is the amount of noise that will be added to the mound, in %.

**Edit Profile** controls the shape of the mound's profile (the shape from the edge of the mound to the highest point). Clicking this button brings up the Remap Altitudes dialog.

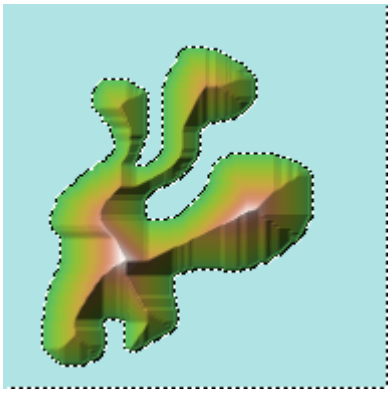
**Operation** is the action to perform with this mound.

**Use Fractal Parameters** indicates if the mound should be created as a linear mound or as a fractal mound. A fractal mound will use the fractal parameters and profile to create the final mound.

**Fractal Parameters** lets you enter the computation parameters for the mound. This dialog is the same one that is used for the Calculate Height Field filter.



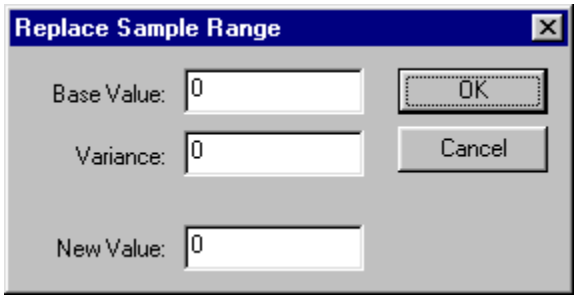
Raw Selection



Default mound

### 3.6.5.6. Replace

The Replace operation allows certain values or ranges of values to be replaced with another value. This operation is useful, for example, to make seas have a minimum depth.




**Base Value** is the desired value to replace.

**Variance** tells by how much the value can vary and still be replaced

**New Value** is the value to replace with.

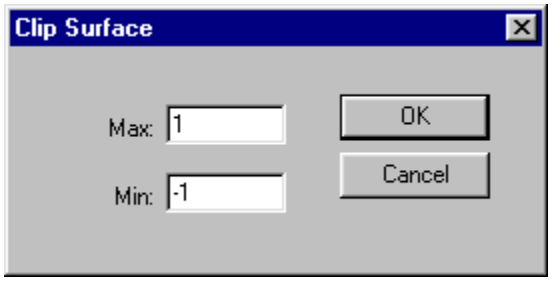
For example, with a **Base Value** of -5000, a **Variance** of 5000, and a **New Value** of 0, the system will replace all values between -10000 and 0 with the value of 0 (effectively removing low-level data).

### 3.6.5.7. Set Value

Asks for a value and sets the entire surface to that value. Setting a selection of the desired shape (and possibly feathering that selection) will create a shape at the height specified.

## 3.6.6. Height Clip

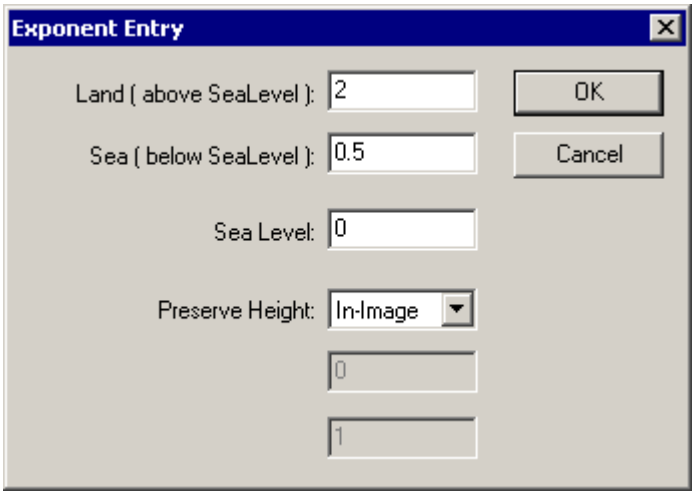
Clips the surface. All values less than min are set to min and all values greater than max are set to max, while values between are left alone.




## 3.6.7. Mathematical

### 3.6.7.1. Exponent

Raises the surface to a power (or more exactly, two powers).




**Land** indicates the exponent that will be applied to points above sea level.

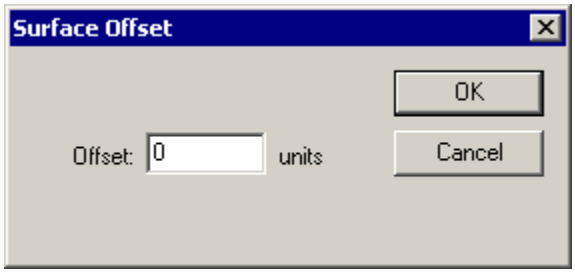
**Sea** indicates the exponent that will be applied to points below sea level.

**Preserve Height** indicates that the height of the highest and lowest points will be the same before and after the operation. If this item is not checked, the values of the highest and lowest points are likely to be greatly modified.

**Sea Level** is the level at which the land/sea exponents will be applied. This parameter is particularly useful when simulating continental shelves, allowing them to be placed below sea level in only a single step.

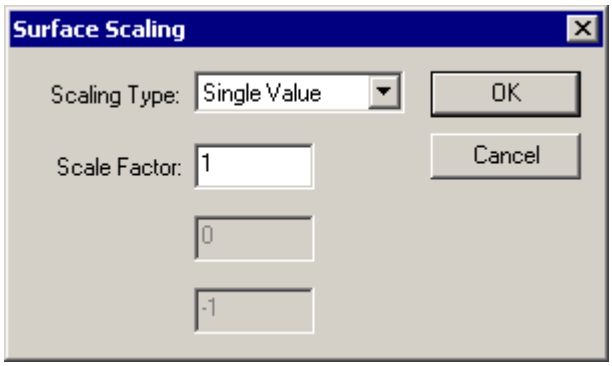
### 3.6.7.2. Offset (Add)

Adds a constant to each sample on the surface.



### 3.6.7.3. Scale (Multiply)

The Scaling command allows the surface to be scaled by a constant value or to a set of absolute heights. It calls the dialog shown below.




**Scaling Type** value has three settings: Single Value, Broken Value, and To Range.

*Single Value* is the simplest. It takes a single parameter (**Scale Factor**) and multiplies every sample on the surface by this value.

*Broken Value* takes three parameters: **High**, **Break**, and **Low**. Each sample with a value greater than **Break** will be multiplied by **High**; each sample with a value less than break will be multiplied by **Low**.

*To Range* takes two parameters: **Highest** and **Lowest**. The entire surface will be scaled and offset such that the highest value on the surface is equal to **Highest** and the lowest value is equal **Lowest**. Useful for quickly forcing the entire surface to a known set of height ranges.

## 3.6.8. Morphological

In image processing, morphological operators are used to adjust the shape and connectivity of an image. Wilbur is not much more than a grayscale image editing program with some peculiar other features tacked on. I didn't implement the opening and closing operators because those are simple Dilate followed by Erode or Erode followed by Dilate.

### 3.6.8.1. Dilate

Implemented as a 3x3 grayscale maximum operator (the largest value in the 9-pixel neighborhood around the current point is used as the new value for that point).

### 3.6.8.2. Erode

Implemented as a 3x3 grayscale minimum operator (the largest value in the 9-pixel neighborhood around the current point is used as the new value for that point).

## 3.6.9. Other

Or, things what I did know know where else I could put them at so they ended up here all alone.

### 3.6.9.1. Cosine Distortion

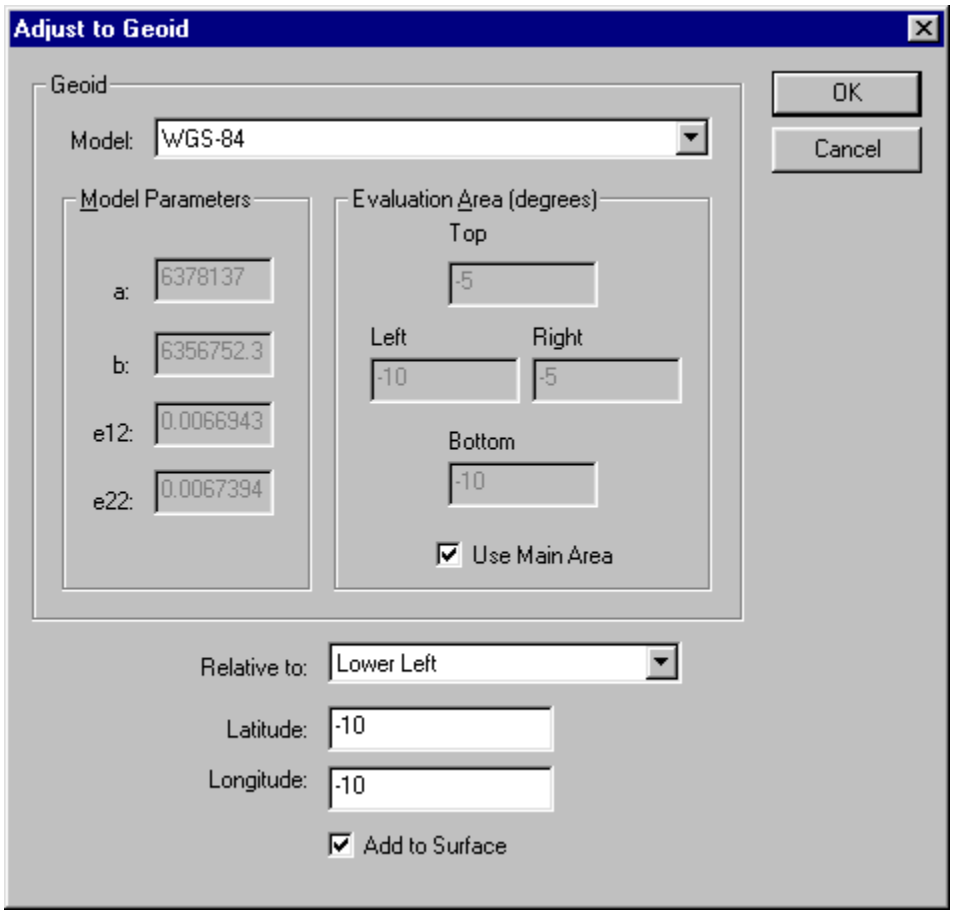
This option will take a flat map and perform a distortion on it that will make it look less squashed in the polar regions when wrapped around a sphere. This was an early attempt to get decent sphere wrapping until it was pointed out that the procedural textures could be directly evaluated on a sphere. It can be interesting, though.

### 3.6.9.2. Fit Geoid

Adjusts the surface to the specified geoid. Because the maps are (typically) defined as a latitude, longitude, and altitude for each point, some programs which deal with the world as a simple height field won't properly display

data. For example, I had a 3D terrain viewer acting as though its terrain were a flat-base height field. When we put the aircraft data into the map (as an XYZ ECEF GPS position converted to a lat/lon and local tangent plane position for display purposes), we found that the plane flew underground a lot. This problem wasn't really noticeable for the 1x1 degree data set, but the 3x3 data set has an error of roughly 2500 vertical meters worst case!

Anyhow, the usage is pretty straightforward. The figure below shows the dialog that shows up when this command is used:




**Model** is the name of the geoid model to use. Some of the more popular ones are NAD-27 and WGS-84.

**a**, **b**, **e12**, and **e22** are the geoid parameters. The data in these windows cannot be entered unless the model is set to *User Model*. Unless you understand the meanings of these values, do not use the user model geoid.

**Top**, **Left**, **Right**, and **Bottom** are the area of the geoid to be evaluated. The data in these windows cannot be changed if the **Use Main Area** checkbox is checked. These values are in degrees.

**Use Main Area** indicates if the edges of the geoid are to be the same as the current edges of the map. If unchecked, any area of the geoid may be used to modify the map data.

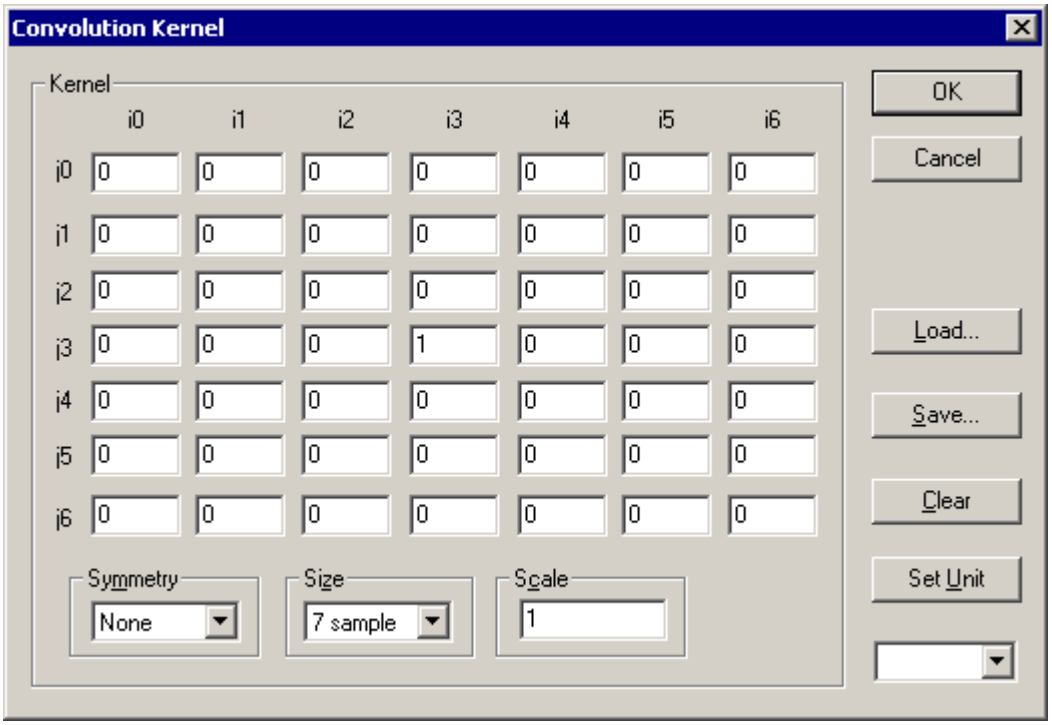
**Relative to** indicates where on the surface the 0,0,0 local tangent point will be located. The current options are the corners, midpoints of the sides, the center of the surface, and an absolute point.

**Latitude** and **Longitude** represent the point to which the *Absolute* **Relative to** value refer.

**Add to Surface** indicates if the geoid data is to be added to (checked) or subtracted from (not checked) the current surface data. This option allows a prior Fit Geoid operation to be undone.

### 3.6.9.3. General Convolution

The general convolution option invokes the Convolution Kernel editing dialog, shown below.



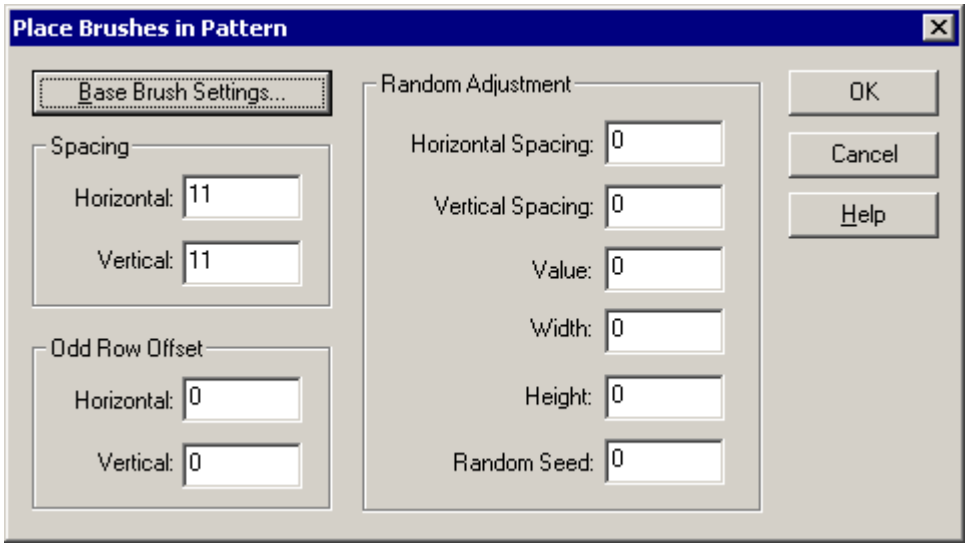

This dialog allows any 3x3, 5x5, or 7x7 convolution kernel to be entered. The Symmetry option is not sticky. Selecting a symmetry sets the current kernel to use the symmetry selected, but does not enforce further changes. The Size option selects the size of the kernel to use: 3x3, 5x5, or 7x7. The entry matrix becomes enabled or disabled to reflect the currently selected size.

The Scale entry holds the scaling factor applied after the convolution occurs. For example, a simple blur uses a matrix composed entirely of 1 values. Using a scale factor of 1 gives a 9x multiplication of the resulting, blurred image. Using a 0.1111111 scale factor (1/9) gives a result that looks substantially like the original image, but is blurred.

The Load and Save options allow a kernel library to be maintained. Clear clears the kernel data and scale factor. The drop list on the right edge will fill the matrix with on of a number of presets. Where applicable, the scaling factor is set according to the currently selected size. For example selecting Blur with a 3x3 surface gives a scale factor of 0.1111111111 (1/9); with a 7x7 matrix, a scale of 0.0204081632 (1/49) is given. The presets currently are blur, sharpen (yeah, I know, actually high-pass), and identity.

### 3.6.9.4. Place Brushes in Pattern

Place Brushes in Pattern will place a brush in a pattern across the surface. This pattern can be a simple rectangular array or a random placement of brushes of different sizes and heights.




**Base Brush Settings** brings up the brush edit dialog to set the basic brush settings.

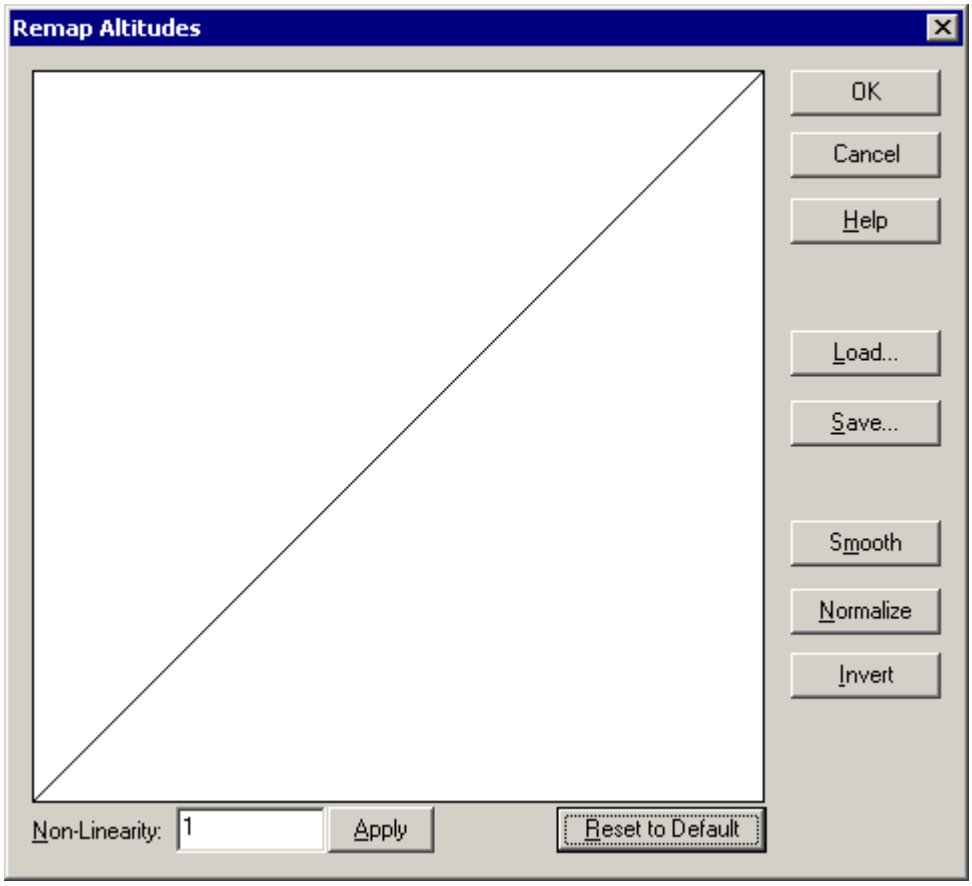
**Spacing** controls how far apart the brush items are placed in the horizontal and vertical directions (values in grid samples).

**Odd Row Offset** controls how the odd rows will be adjusted from their place in the grid (in grid samples). Setting this value to half the horizontal spacing will give you a nice hexagonal array instead of purely rectangular one.

The **Random Adjustment** group controls the amount of randomness that will be added to the appropriate brush parameters. **Random Seed** controls the sequence of random numbers generated (for repeatability).

### 3.6.9.5. Remap Altitudes

The remap altitudes operation changes the current terrain altitudes into new ones. You have the option of performing no action (the default) or mapping in other ways, including drawing into the main area to set new information.




The main drawing area is both a preview of the current mapping as well as a place to draw your own mapping. Click and drag with the left mouse button to draw a new curve.

**Non-Linearity** is the amount of non-linear scaling applied to the current mapping. A value of 1 will have no effect; values less than 1 will make a convex (bulge up) result; values greater than 1 will make a concave (bulge down) result. **Apply** will apply the non-linear scaling factor to the map.

**Reset to Default** will reset the map to its default linear configuration.

**Load** loads a remap file from disk.

**Save** saves the current mapping to disk for future use.

**Smooth** removes some sharp edges from the map. Multiple uses of this tool will naturally lead to smoother results.

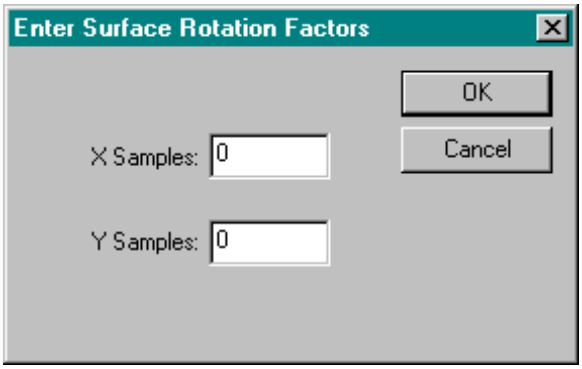
**Normalize** will ensure that the map covers the entire range from 0 to 1 vertically. Application of the smooth tool or drawing to the edges can result in a map that does not cover the full vertical range. This tool fixes that problem.

**Invert** inverts the map vertically.

### 3.6.9.6. Toroidal Rotation

This operation refers to a toroidal rotation, not a planar rotation. What that means is that things rotated in Y appear to roll and wrap from top-to-bottom. For X, it's left-to-right. This option is useful for making "seamless" maps. You rotate the image by an amount to paint or process the erstwhile edges. A blur operation from the paintbrush can help blend the edges together, giving a seamless result.

Anyway, here's a picture of the dialog:




**X Samples** and **Y Samples** are the number of columns/rows to rotate the surface by. Use positive values for rotation in one direction, negative values for rotation the other direction, or 0 for no rotation in that axis.

## 3.6.10. Sharpen

### 3.6.10.1. Some (3x3)

Uses the general convolution operation with a 3x3 sharpening kernel.

### 3.6.10.2. More (5x5)

Uses the general convolution operation with a 5x5 sharpening kernel.

### 3.6.10.3. Lots (7x7)

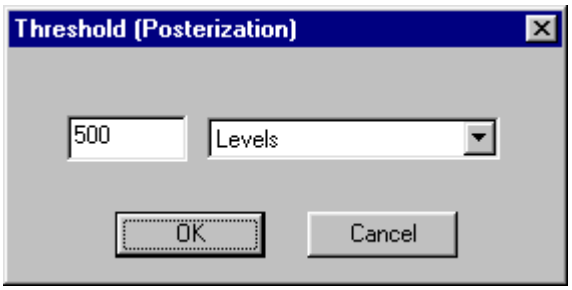
Uses the general convolution operation with a 7x7 sharpening kernel.

### 3.6.10.4. Unsharp Mask

Performs a gaussian blur and then subtracts the blurred image from the base image. That's what an unsharp mask does, honest! It dates back to old photographer tricks involving developing one image out of focus and then stacking it with the original negative for printing. I think I did it wrong, though.

## 3.6.11. Threshold

Also known as "posterizing". Converts the nice smooth surface into a series of obviously discrete steps. This command calls the dialog shown below:




The value represents either the number of levels in the final result or the size of each step, depending on the setting of the drop list. The drop list has two settings: *Levels* and *Meters per Interval*.

When *Levels* is chosen, the surface will be modified so that there are the number of discrete levels is equal to the number of levels indicated in the dialog, with as much distance between each step as is required to get that number.

In contrast, when the *Meters per Interval* is chosen, the surface will be modified to give an arbitrary number of levels, each separated by a magnitude equal to exactly that given in the dialog.

## *3.7. View*

### 3.7.1. Status Bar

Shows or hides the status bar at the bottom of the screen. Hiding the status bar gives slightly more room onscreen. If the status bar is visible, this menu item will be checked.

### 3.7.2. Toolbar

Shows or hides the toolbar onscreen. Hiding the toolbar gives slightly more room onscreen. If the toolbar is visible, this menu item will be checked. The toolbar may also be undocked from the edge of the window by dragging it away; it may be redocked to any edge of the window by dragging it back.

### 3.7.3. Painting Tools

Shows or hides the painting tools toolbar onscreen. If this toolbar is visible, the menu item will be checked. The toolbar may also be undocked from the edge of the window by dragging it away; it may be redocked to any edge of the window by dragging it back.

### 3.7.4. Zoom

The current facilities for zooming in on an image are embodied in this menu. There are no fancy toolbars, no click-and-drag zooms, nothing really fun to use. Just generic, ugly menu items. The zoom features don't necessarily maintain the correct location when zooming, so you may end up in unfamiliar territory after the zoom.

The zoom menu is used to change the apparent scale of the image. It is very useful when an overview is desired or when "fine detail" needs to be seen. When an image is shown larger than the actual pixel size, the individual pixels are replicated with no sampling, giving the typical mosaic pattern. When zoomed out, no area sampling takes place, just decimation of the image, so data will be aliased (lines may appear to be broken, etc.)

#### 3.7.4.1. Zoom Full Size

Shows the image at 1 sample = 1 pixel.

#### 3.7.4.2. Zoom In 2X

Changes the current displayed scale by a factor of two. Half as much of the image is visible, but at the same amount of screen space.

#### 3.7.4.3. Zoom Out 2X

Changes the current displayed scale by a factor of one half. Twice as much of the image is visible, but at the same amount of screen space.

#### 3.7.4.4. Arbitrary

This operation asks for a scale factor. Factors larger than 1 will make the viewed image appear larger onscreen, while values smaller than 1 will make the view appear smaller. A value of 1 will show the image at full size.

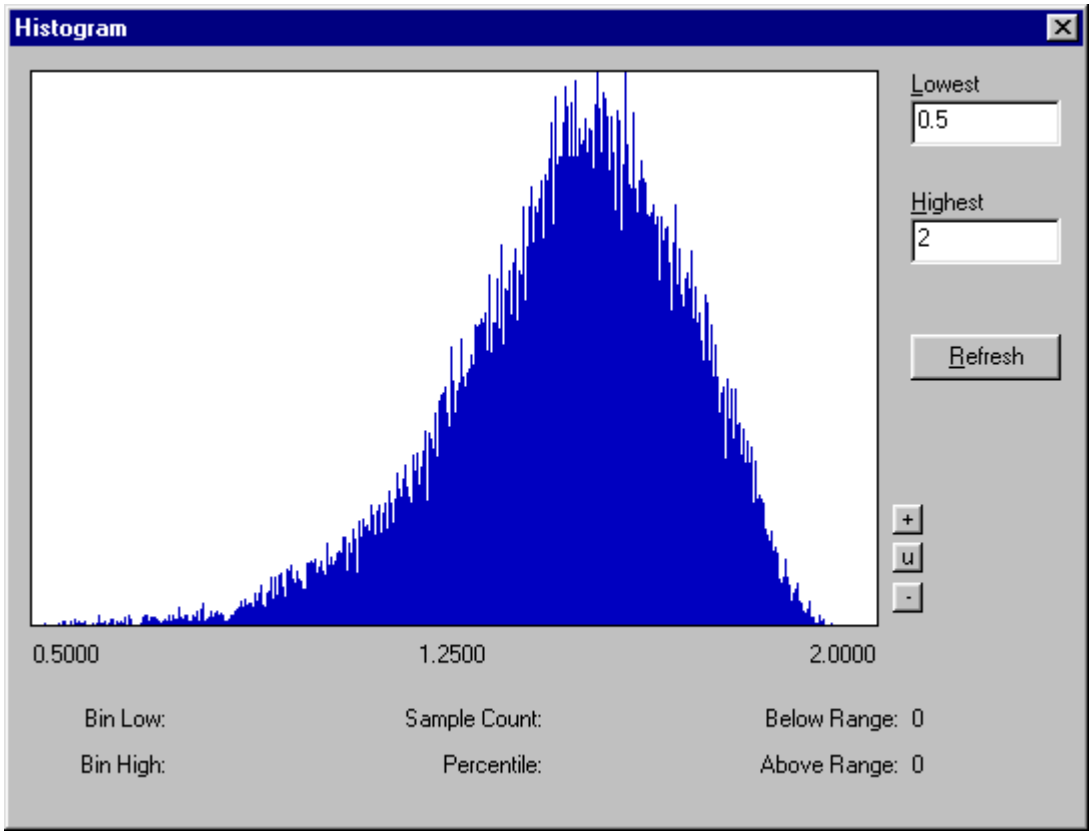
### 3.7.5. Refresh Display

Causes the image in the display to repaint. This option is useful on those occasions where another program has corrupted the display and it needs to be redrawn. This particular problem shouldn’t happen anymore, but it never hurts to keep it around.

## *3.8. Window*

### 3.8.1. Histogram

Sometimes you may want to explore the statistics of your surface. The easiest way to do that sort of analysis is to count the number of samples that fall into a number of bins between a high and low value. To assist you in you analysis (and to find out what some of the screwier routines are doing), the Histogram dialog is provided. An example is shown below:




**Lowest** and **Highest** control the lowest and highest displayed elevations, respectively. The defaults for these entries are the highest and lowest points on the surface. By changing the values, a histogram of selected range values may be views. **Below Range** and **Above Range** show the number of pixels above and below the displayed histogram range, respectively.

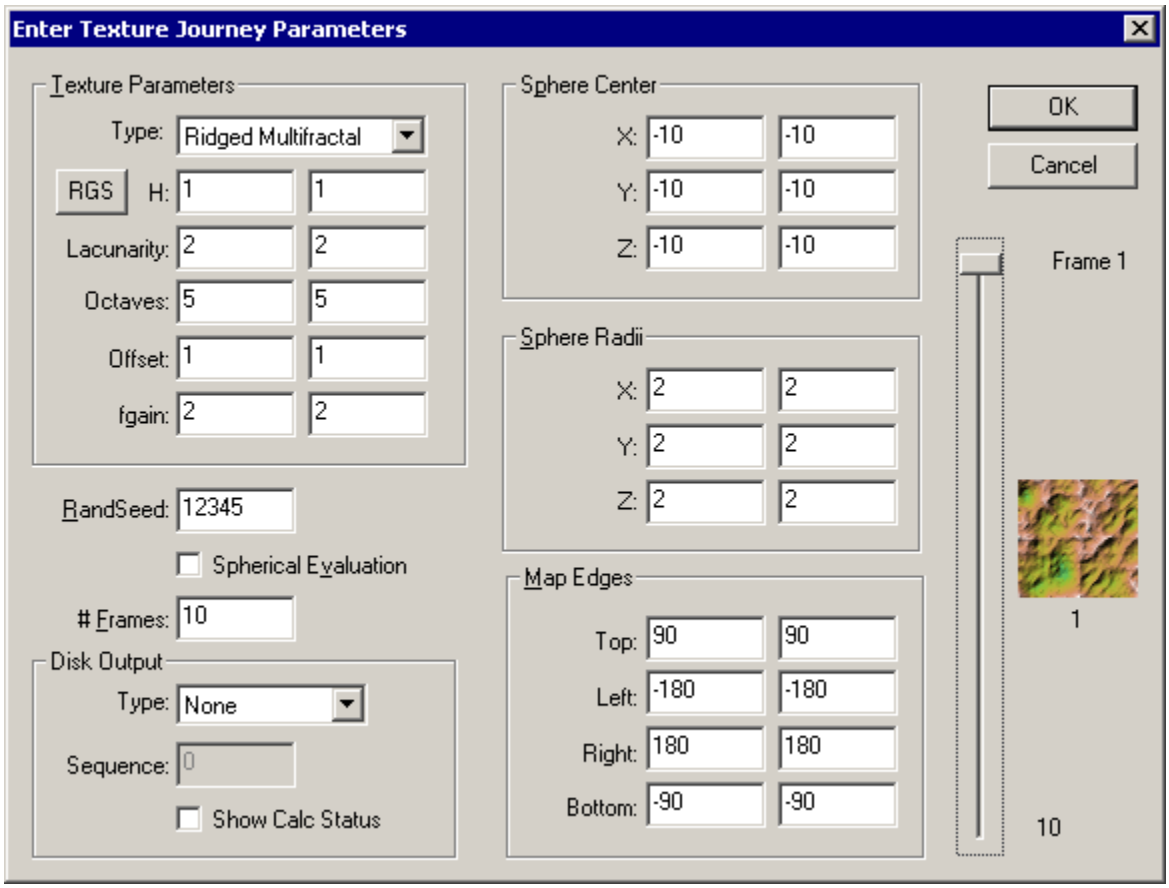
The large white area shows the actual computed histogram. The basic size of the histogram is 420 bins, spread evenly from lowest to highest. The histogram is scaled vertically by default so that the largest element completely covers the graph vertically and all others are scaled relative to that one.

**Bin Low**, **Bin High**, and **Sample Count** show the data for the bin currently under the cursor. **Bin Low** and **Bin High** values show the lower and upper bounds for the bin under the cursor. The **Sample Count** value shows the number of samples that fell into that bin (between **Bin Low** and **Bin High**). Moving the mouse cursor across the histogram displays the data for the relevant bin. **Percentile** is the percentage of the total samples found below the current elevation indicated by the cursor position.

The **+** button increases the vertical scale by a factor of 2, the **-** button decreases the scale by a factor of 2 and the **u** button sets the vertical scaling to its default value (largest bin taking full height).

## 3.8.2. Journey Through Texture Space

There comes a point where it just takes too long to find a good terrain by pecking numbers into the computer. To speed up the process, the Journey Through Texture Space was added. Basically, this feature calculates a procedural texture surface just like the Calculate Height Field command, but it lets each parameter be interpolated between two values. Unlike the Calculate Height field, the data also includes the edges of the map. The dialog is shown below:




The only surface types supported are the procedural texture types described under the Calculate Height Field section except for the plasma and math function types.

The **RGS** button loads the "from" and "to" values with the current system parameters. What that means is if you find a surface you like, you can use it as the starting point for a little excursion through texture space.

The ***Texture*** parameters (and position, size, and Map Edges) have two edit boxes for each parameter. The lefthand box of each set is the starting ("from") parameter and the right hand or bottom is the ending ("to") value.

**# Frames** shows the number of frames that will be processed.

**RandSeed** is the random number seed used to generate the surface.

**Spherical Evaluation**, if checked, makes the surface calculate data over a sphere instead of a plane. In spherical mode, Position is the location in 3space of the center of an ellipsoid, Size indicates the three radii, and Map Edges gives the section of the sphere to evaluate over. The Map Edges are available for data entry *only* if Spherical Evaluation is checked.

**BMP Seq** is the index number of the first file to set. If this value is greater than or equal to 0, then the entire frame sequence will be saved to disk in the current directory. All file names will be of the form PICXXXXX where the XXXXX is replaced by the index number.

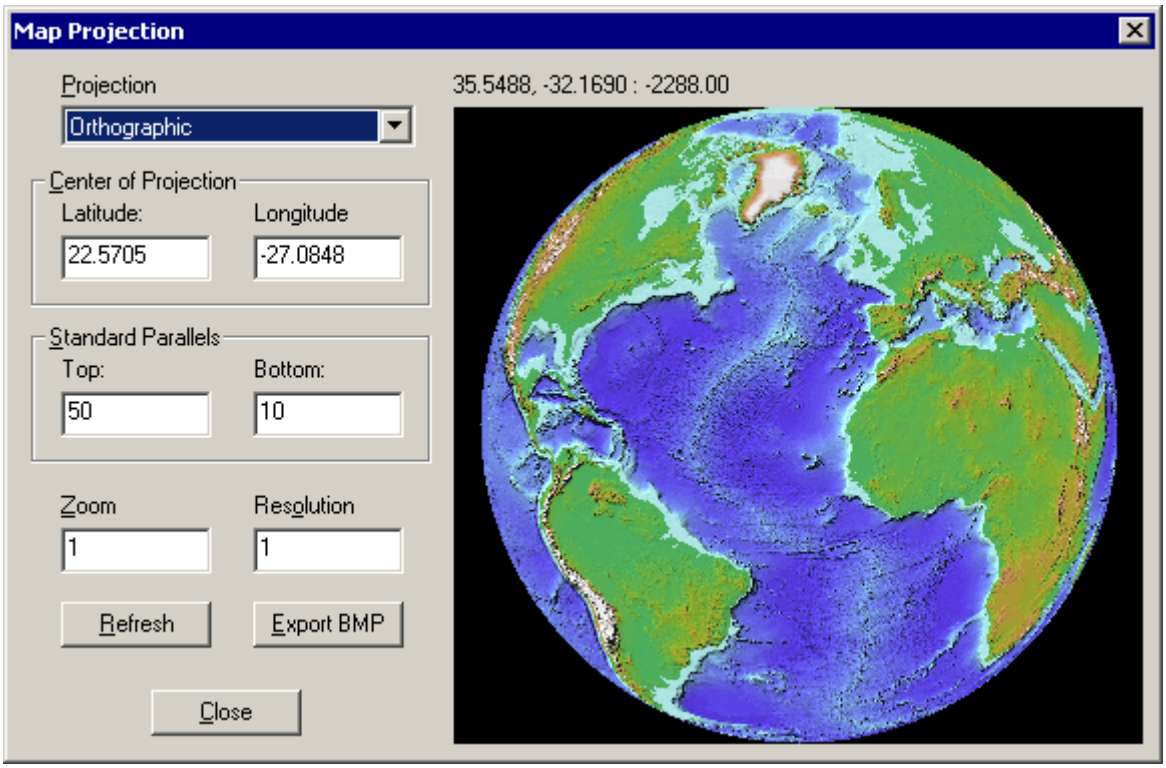
**Show Calc Status** indicates if the calculation status dialogs should be show during processing. For very small surfaces ("real-time" operation), it is best to turn off this item because the display flickers as the progress dialogs

come up and go down. Calculating a 128x64 surface with the above parameters gives about 2 frames per second on my machine and the flickering get downright annoying.

The slider and the rest of the stuff on the right hand edge of the dialog show a tiny preview image of what will be created for each frame in the journey.

### 3.8.3. Map Projection

The flat display shown by Wilbur as its default view will not exactly give a very good idea of how things fit together, especially at the poles. The stretching gets rather extreme at the poles. Most maps in use today do not use this projection (called the Equirectangular or Plate Caree projection). To support other projections, the Map Projection dialog (shown below) is provided.




**Projection** is a drop list containing the type of projection shown in the window at left. The list of projections and an example of each is shown below.

The center of projection will be shown in the **Latitude** and **Longitude** areas at the right of the screen. To move the center of projection, just move the mouse over the map, click and drag. The direction might not be exactly what you expect and varies with the type of projection. During the drag, the map resolution will be reduced to get an interactive display rate. Note that there are currently no facilities for moving the map area around except at the center of projection.

***Standard Parallels*** are the parallels used for the conic projections. Not applicable to any other projections.

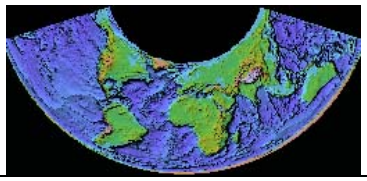
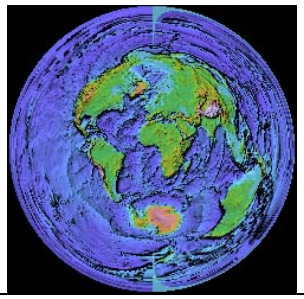
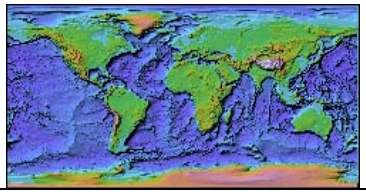
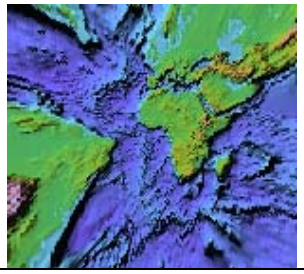
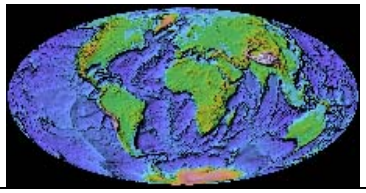
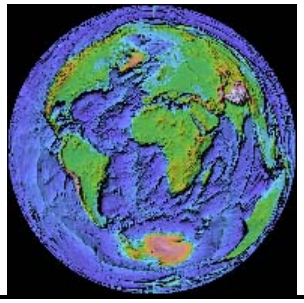
**Zoom** is the zoom factor for the image. A value of 1 shows a 1:1 view of projection, which may not be the whole world for some projections.

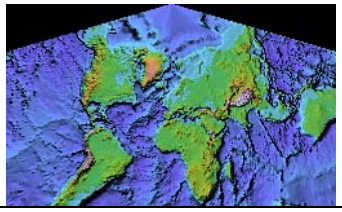
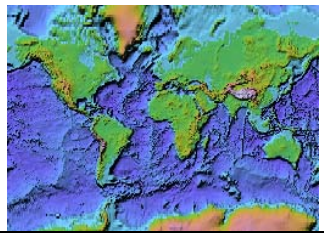
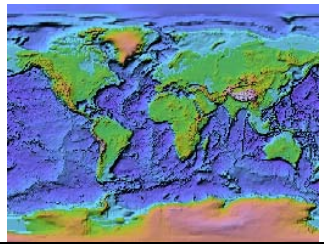
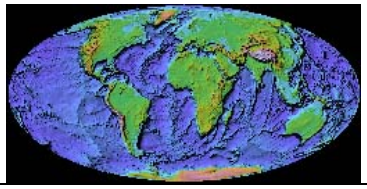
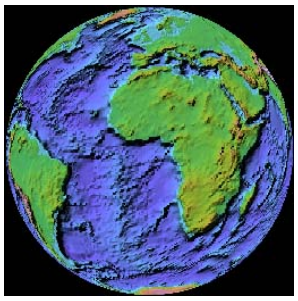
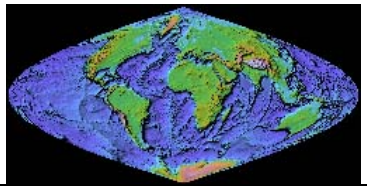
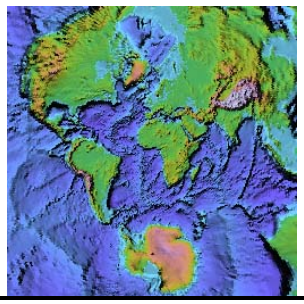
**Resolution** provides a tradeoff between quality of the display and the time it takes to redraw. On a Pentium 166, a Resolution value of 1 takes a little over a second to draw; a value of 4 allows interactive update rates (at the cost of a slightly chunky display).

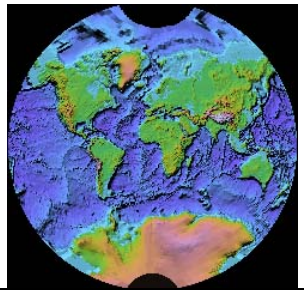
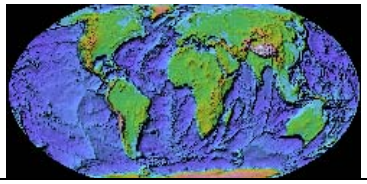
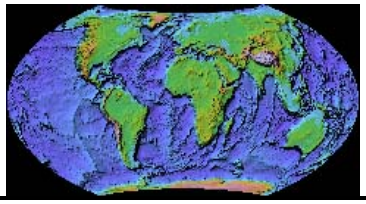
**Refresh** redraws the display using the current parameters. If the mouse has been used to drag a view around, the parameters shown may not exactly match the view displayed. Hit this button to be sure of repeatability.

**Export BMP** allows the current view to be output to a 24-bit color BMP file of any desired size.

Projection Summary

| Name | Sample Whole-World Image | Features | Suggested Zoom |
|---|---|---|---|
| Albers Equal-Area Conic |  | | 0.333 |
| Azimuthal Equidistant |  | | 0.3 |
| Equidistant Conic | ?? | Problems getting this one to work properly. | ?? |
| Equirectangular |  | Simplest projection, equal pixel distance between lines of latitude and longitude. This is the projection used by Wilbur internally. Extreme distortion in polar areas (the poles become lines). | 0.3 |
| Gnomonic |  | | N/A (0.35) |
| Hammer |  | | 0.35 |
| Lambert Azimuthal Equal-Area |  | | 0.5 |

| | | | |
|---|---|---|---|
| Lambert Conformal Conic |  | | N/A (0.35) |
| Mercator |  | | N/A (0.3) |
| Miller Cylindrical |  | | 0.3 |
| Mollweide |  | | 0.3 |
| Orthographic |  | Hemispherical view. Appearance as if the globe were being viewed form space.<br>Also has a bug that causes less than whole-world areas to be shown on the opposite side of the equator. The workaround for this problem is to reverse the sign and positions of the top and bottom on the map e.g. a map has top=36, bottom=35; the correct projection will come when top=-35 and bottom=-36. Sorry about this problem. | 1.0<br>64.0 for a 1x1 degree area centered in the view. |
| Sinusoidal |  | Equal-Area. Distorts the shapes but preserves their areas. | 0.3 |
| Stereographic |  | | N/A (0.3) |

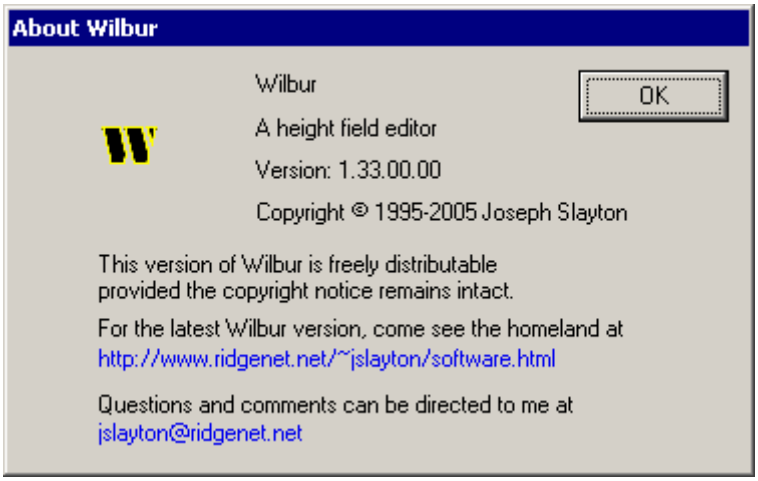
| | | | |
|---|---|---|---|
| Van Der Grinten |  | | 0.3 |
| Wagner IV |  | | 0.3 |
| Wagner VII |  | | 0.3 |

### *3.9. Help*

The help menu contains the program help facility. There's currently not much help available, just an about box. That may change in a future version, but don't hold your breath.

#### 3.9.1. About

As you might imagine, this command shows the About dialog box, shown below:




It’s generic and uninteresting, eh?

## 4. System toolbar

In order from the left, the buttons are: New File, Open File, Save File As, Undo, Redo, Pan Image, Zoom Image, Select Rectangle, Select Freehand, Custom Paint Tool, Custom Clone Tool, Flood Fill Tool, Gradient Tool, Line Tool, Relight, and About. These items are discussed in further detail below. Note that these buttons replicate exactly certain menu items. To help you remember which menu items, those items will have the little pictures next to them.

This toolbar implements tooltips; for a refresher about the use of a button, let the mouse cursor rest over the button for a second or so; a small window will appear that describes the function of that tool.

The tool buttons (pan, zoom, the selection tools and painting tools) latch when used. That is to say that they remain pressed after the tool selection is made. The cursor will also change to reflect the current tool. These tools may be deselected by selecting the tool again (which will cause the button to stop being pressed).

## *4.1. File Tools*

The File Tools are shortcuts to the most commonly used File menu items. Each tool in this section exactly replicates the function shown on the File menu.

### 4.1.1. New

Selecting the New toolbar item has the same effect as the selecting the File → New menu item (see 3.1.1).

### 4.1.2. Open

Selecting the Open toolbar item has the same effect as the selecting the File → Open menu item (see 3.1.2).

### 4.1.3. Save As

Selecting the Save As toolbar item has the same effect as the selecting the File → Save As menu item (see 3.1.3).

## *4.2. Edit Tools*

### 4.2.1. Undo

Selecting this tool has the same effect as selecting the Edit→Undo menu item (see 3.2.1).

### 4.2.2. Redo

Selecting this tool has the same effect as selecting the Edit→Redo menu item (see 3.2.2).

## *4.3. View Appearance Tools*

### 4.3.1. Pan

The Pan tool provides a direct way to manipulate the viewed area of the map. It is mostly useful in the context where the map boundaries are outside of the displayed screen area and the scroll bars are displayed.

When this tool is selected, a cursor resembling an open hand is shown. Move this cursor over the map and click the primary mouse button. The cursor will change to a closed hand and moving the mouse with the hand closed will drag the displayed map area around the physical map. Release the button to stop dragging the viewed area.

### 4.3.2. Zoom

The Zoom tool provides quick access to zooming in and out on the map. When this tool is selected, the cursor will appear as a magnifying glass with a '+' in the center. Pressing the Shift key will change this symbol in the center to a '-'.

Clicking the primary button when the cursor shows a '+' will zoom in by a factor of 2; Clicking the primary button when a '-' is showing will zoom out by a factor of 2. These functions are identical to the View→Zoom→In and View→Zoom→Out menu items, respectively.

Note that the location of the mouse does not necessarily affect the location zoomed in to. Also, a zoom box operation (selecting an area to zoom into) is also not yet implemented.

## *4.4. Selection Tools*

### 4.4.1. Rectangle Selection

Selecting this tool has the same effect as selecting the Select→Rectangular Selection Tool menu item (see 3.5.5).

### 4.4.2. Freehand Selection

Selecting this tool has the same effect as selecting the Surface→Flood Fill Tool menu item (see 3.5.5.4).

## *4.5. Painting Tools*

### 4.5.1. Paint

Selecting this tool has the same effect as selecting the Surface→Custom Paint menu item (see 3.3.6).

### 4.5.2. Clone

Selecting this tool has the same effect as selecting the Surface→Custom Clone menu item (see 3.3.7).

### 4.5.3. Flood Fill

Selecting this tool has the same effect as selecting the Surface→Flood Fill Tool menu item (see 3.3.7).

### 4.5.4. Gradient

Selecting this tool has the same effect as selecting the Surface→Gradient Tool menu item (see 3.3.7).

### 4.5.5. Line

Selecting this tool has the same effect as selecting the Surface→Line Tool menu item (see 3.3.7).

## *4.6. Misc Tools*

### 4.6.1. Relight

The relight tool scans the current surface for the minimum/maximum values, then recomputes the lighting with those parameters. This tool is useful if painting with the raise or dig tools, because using it will use the full dynamic color range of the lighting model.

## *4.7. Help Tools*

### 4.7.1. About

Shows the system about box (see 3.9.1)

# 5. Painting toolbar

Most people never seemed to grasp the uses that the custom paintbrush tool could be put to; it's not overly intuitive. To solve this problem, the painting toolbar was born. It provides a quick-and-easy access point to the basic functionality of the painting tool. Everything on this toolbar can be done from the Edit Paintbrush dialog, but it's a little bit easier to use in this form.

## 5.1. Basic Tools

The basic operation tools on the painting toolbar are colored yellowish-brown. These tools are all aspects of the basic painting tool found on the main toolbar. In order from left to right, they are Paint, Dig, Raise, Blur, Sharpen, Noise, and Erode.

### 5.1.1. Paint

The Paint tool moves the surface covered by the brush under the mouse towards the value set using the intensity tool.

### 5.1.2. Dig

The Dig tool lowers the surface covered by the brush under the mouse by the value set using the intensity tool.

### 5.1.3. Raise

The Raise tool lifts the surface covered by the brush under the mouse by the value set using the intensity tool.

### 5.1.4. Blur

The Blur tool softens the surface covered by the brush. The intensity value has no effect on this operation, only the brush size and shape.

### 5.1.5. Sharpen

The Sharpen tool increases the contrast along edges for the surface covered by the brush. The intensity value has no effect on this operation, only the brush size and shape.

### 5.1.6. Noise

The Noise tool adds a random value to the surface covered by the brush under the mouse. This noise has a maximum value of +/- the value set using the intensity tool.

## 5.2. Intensity and Shape tools

The intensity and shape tools control the form of the brush. They are color-coded gray in the toolbar. In order from left to right they are Intensity, Square, and Round.

### 5.3. Intensity

Selecting this tool will bring up a dialog that allows the intensity and pointiness of the tool to be controlled. Intensity is the amount of effect that the tool will have (or the desired elevation in the case of the basic painting tool). The pointiness parameter is the same as the bias parameter in the Custom editing dialog. This value ranges from 0.0 (very rounded on top with steep sides), through 0.5 (a cone shape from the edges to the center), and on to 1.0 (very horizontal edges with a high central peak).

### 5.4. Square

The Square button changes the shape of the tool to a square that doesn't feather out the operation into the surface. Performing a Raise operation will give a square block of the desired altitude, for example.

### 5.5. Round

The Round button will changes the shape of the tool to a round tool that feathers out nicely to the edge. The shape of the feathering is controlled by the pointiness value entered in the Intensity tool, while the amount of effect for the tool is controlled by the intensity value.

### *5.6. Size tools*

The Size tools control the size of the brush and are color-coded greeen. From left to right the sizes are 51x51, 41x41, 31x31, 21x21, 11x11, and 5x5. The shape and intensity of the tool are controlled by the Intensity and Shape tools.

### *5.7. Custom tool*

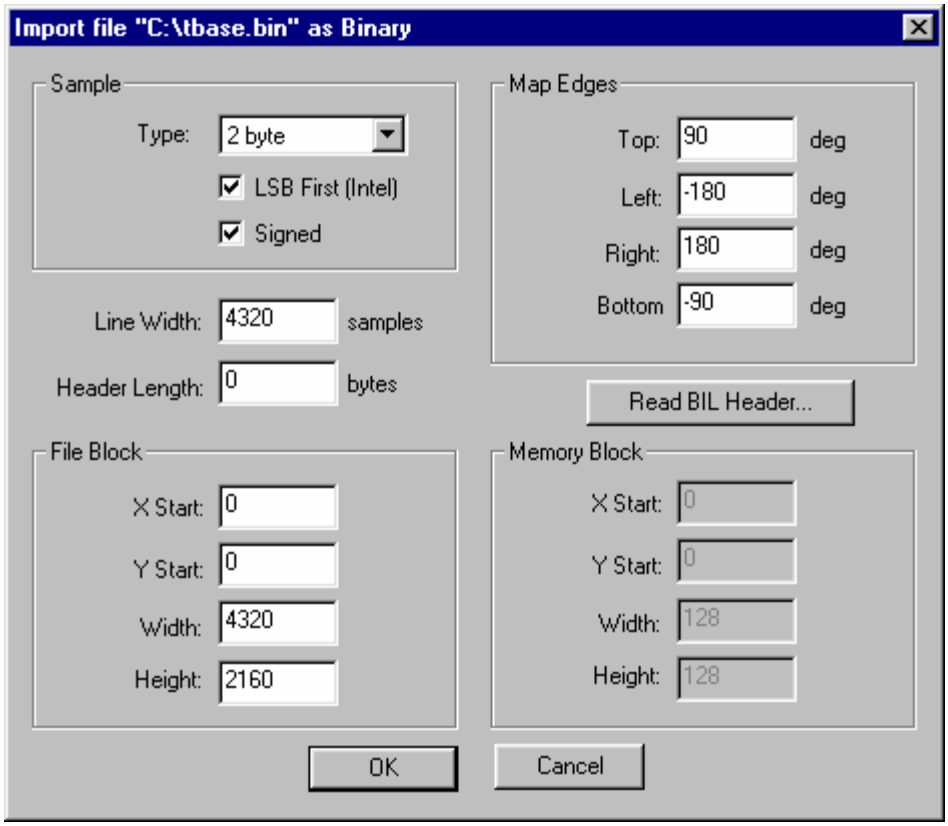
The custom tool allows all options of the paint tool to be controlled. It is colored blue. This button has the same effect as the Surface→Edit Paintbrush menu item. See that section for details.

## 6. Real-World Data Sets

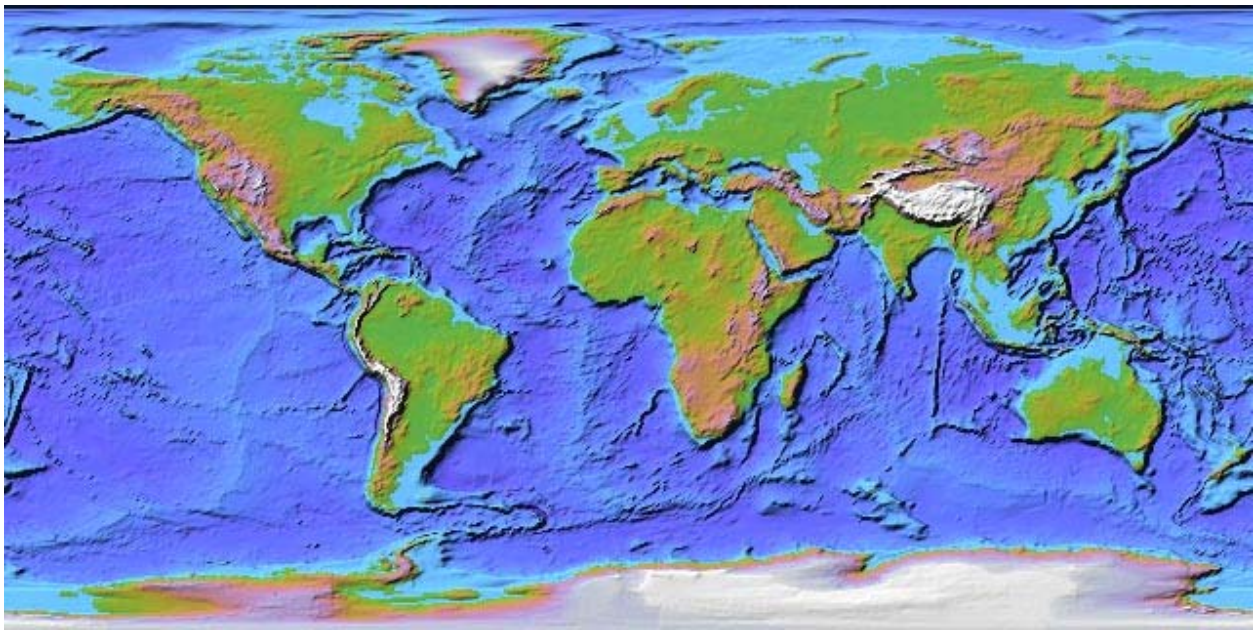
Wilbur generates synthetic data sets, but it also handles real-world raster data sets pretty well. There are good data sets available on the Internet; some of them, however, are rather large. The best three (the ones I usually use anyway) for making world and local maps are described below:

### *6.1. TerrainBase*

TerrainBase is a 10 km per sample data set of the whole world. It's about 18 MB to download in uncompressed form, but it's well worth the effort. To open the entire file, use the "Binary Surface" file type with the parameters shown below:



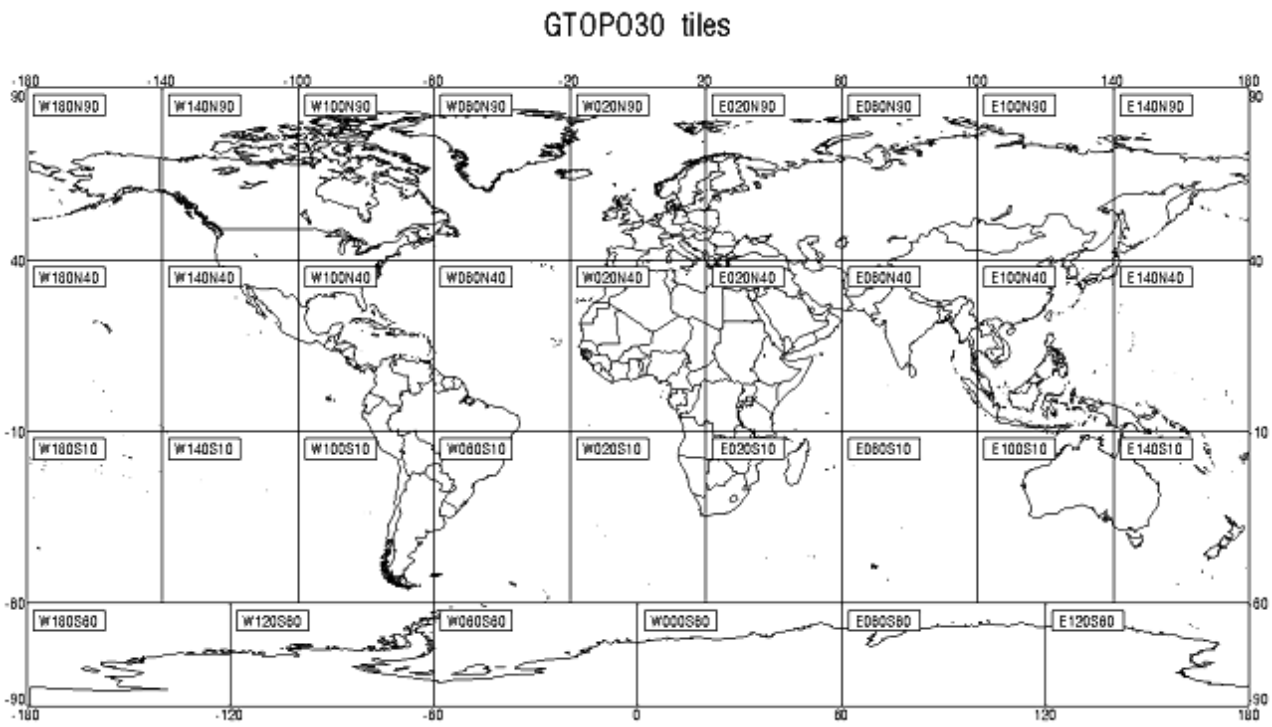
A simple representation of the data set (no adjustment for land parts below sea level) is shown below: This image was downsampled from the full 4320x2160 data set to fit into a 640x320 image in order to keep the size of this document reasonable.



The TerrainBase CD-ROM download area is located at http://www.ngdc.noaa.gov/seg/fliers/se-1104.html. Pick the "GLOBAL LAND AND OCEANS" data set.

## 6.2. GTOPO30

GTOPO30 the USGS has all of the world's land masses at roughly 1 km resolution. The entire data set is very impressive. Below is the image on the main GTOPO30 page at http://edcwww.cr.usgs.gov/landdaac/gtopo30/gtopo30.html showing the tiles in the data set.




Be warned, though, this thing is *huge*. Most of the blocks on the diagram are 4800x6000 pixels, or about 57.6 MB on disk in uncompressed format. Each. There are 27 of them (plus Antarctica). The next tutorial has some good images of some of the data from this set. The only real downside (other than the 275 MB compressed download or 5-CD set from USGS) is that there's not any seafloor topography.

To use a data block from the set, download the data. Uncompress (gzip) and untar the data file. You only need to keep the .DEM file and .HDR file to use with Wilbur. The rest of the data is nice, but Wilbur doesn't care about it.

### 6.3. USGS United States DEMs

The USGS also has data for the United States at roughly 90 meter resolution online. A good starting point is http://edcwww.cr.usgs.gov/glis/hyper/guide/1_dgr_demfig/index1m.html, a page that starts out with a clickable map of the US used to select the parts you want to get. The data files are pretty good sized (9 MB uncompressed and up to 2 MB compressed). You'll need a program that can handle gzip-format files to get the compressed files. To use the files with Wilbur, just uncompress the files and load them in using the "USGS DEM" format. Saving as "Muse DTED" will result in a file that loads lots faster next time and also has the correct positioning information saved in it.

### 6.4. USGS SDTS DEMs

The USGS also has many 7.5 minute DEMs online, but only in SDTS format. Wilbur currently does not support this file format.

## 7. Tutorials

The tutorials have been moved to a separate file, available from the same source as this documentation file. If the file wasn't where you got this file, try the main Wilbur page at http://www.ridgenet.net/~jslayton/software.html and look on the download pages.

# 8. Credits and history:

This software has been cobbled together from bits of other software and a lot of work on my part to get it to work. Many of the file I/O routines came from various places, as did the DTED import routine and lighted color algorithm.

It all started several years ago when I was playing with a 16-bit DOS plasma fractal generator written in C called BOGUSSAT (where some of the plasma terrain generator and color lighting came from). It was pretty, but I wanted more. I added various processing options and some file I/O. I ported it to a 32-bit DOS C compiler to get bigger surfaces. I added SuperVGA output options. I played with it a lot.

Then a customer I work for saw some of the output. Eventually, I got a job to process the coverage and power levels for FM transmitters around the area. They got me some terrain data and some routines to read that data (that's where parts of the DTED import and Muse I/O come from) and paid me to port the whole thing to Windows 3.1. That meant a shift back from 32-bit DOS to 16-bit Windows and a shift from command-line driven to menu-driven.

Then I got a Visual C++ 2.2 package and ported the whole thing to 32-bit windows.

Over time more processing has been added and various I/O options other bits have been grafted onto the software. New synthetic terrain types were added. New file I/O options were supported (TARGA surfaces and images, for example). Some parts were removed (line of sight calculations, rf emitter and site managers, for example). That history brings us up to version 1.05.

After getting a request to make the software deal with MATLAB more effectively, I read through the HFLAB software source, which supports the MATLAB 1.0 matrix file format. I deduced the file format from the code and then used that file format to write code to read and write the data sets. I don't have MATLAB, so I'm hoping I did it correctly.

# 9. Versions

| | | |
|---|---|---|
| 1.05 | Date | First public release |
| 1.06 | | Added 3D DXF export and this document file. |
| | | Fixed a bug that caused program crashes when a math surface was calculated, followed by a procedural texture. |
| | | Added the curval parser entry. |
| | | Recovered lots of palette entries that were no longer being used for coverage overlays and fixed everywhere that used them. |
| | | Added a history list to the expression data entry location. |
| 1.07 | | Fixed a significant bug in the convolution routines that smeared all data diagonally. |
| | | All data was smeared diagonally; it's been fixed. Added fbm and noise to function list. |
| | | Added RAW triangle output. |
| 1.08 | | Added very primitive painting capabilities. Lets you get the effect of canyons, etc. by drawing on the surface. |
| | | The TIFF file I/O support didn't work, so it was excised. |
| 1.09 | | Added binary insert. Inserts a binary image into part of the file in memory. Makes it easier to add relevant bits of terrain. |
| | | Added an option to the DXF/Raw output that lets the number of decimal places be specified. |
| | | Fixed height coloring bug added in V1.09. |
| | | Got rid of 16-bit legacy that limited resize to 2048x2048 samples. |
| 1.10 | | Added painting options. |
| | | Added a semi-useful documentation file. |
| | | Cleaned up lots of little bits that didn't work quite right. |
| 1.11 | | Fixed problem with painting that caused only parts of the brush to be drawn when within the |

| | |
|---|---|
| | brush radius of the edge. |
| | Cleaned up unused dialogs and resource identifiers. |
| | Fixed long-standing palette allocation bug. |
| | Made usability adjustments to the palette dialog (added colors/levels slider and removed unused stuff). |
| | Fixed lighting problem that caused the upper and right edge of the map to not be calculated in lighted modes. |
| | Fixed color overflows caused by non-integral palette calculation. |
| | Added new brush modes to the painting tool (notably convolution as a real-time operation) |
| | Adjusted the "pure height map" option to use a grayscale palette from 0 to 255 to match the actual height info. The palette appears screwed up onscreen, but that's because the Windows Palette Manager allows only 236 colors instead of 256. |
| | Added some simple presets to the convolution kernel edit dialog. |
| 1.11a | Added a lighting direction visualization display |
| | Added hex grid overlays |
| | Fixed some crashes with convolution. |
| 1.11b | Fixed bug with saving files of differing type but same file extension (specifically Fractint TARGA files). |
| 1.11c | Added 24-bit TARGA SuperImage output |
| 1.11d | Fixed some palette bugs |
| | Gamma works properly with the 24-bit file output. |
| 1.11e | Reorganized menus |
| | Fixes some bugs |
| | Added cosine distortion process |
| | Added rectangular grid |
| | Added contrast enhancement options to hex and rectangular grid overlays. |
| 1.11f | Reads 8-bit BMP files as height fields |
| | Added spherical procedural texture evaluation to make spherical world maps easier. |
| | Added "Scale To" and "Broken Scale" to scaling dialog |
| 1.11g | Made the exponential operation a single-pass operation (much faster for large surfaces). |
| | Removed Clear, Unitize, and Normalize point processes because their function is provided by the Height Clip and Scale functions. |
| | Added a histogram dialog to help find those pesky high-octave-value outliers. |
| | Fixed the sea color problem (the sea color was constant and didn't respond to setting in the palette). |
| | The RandSeed parameter wasn't being used during procedural texture calculations. Fixed. |
| | Printing works (at printer resolution, anyway), no randseed for procedural textures, and the printing finally sorta works (apologies to those using the 1.11f version). |
| 1.11h | Added more brush controls including inclusive and exclusive operation ranges. |
| | Added low and high values to histogram dialog to enable histograms of arbitrary portions of the height range. |
| | Finally redid the documentation. It's not great, but at least it's not 9 versions (almost 2 months!) out of date. |
| 1.12 | Fixed a bug with the 8-bit BMP surface input that only read part of a non-square image. |

| | |
|---|---|
| | Fixed the convolution brush. Wasn't working. |
| | Fixed beta-spline interpolation. |
| | Changed interpolation dialog to only use bias and tension if beta-spline type |
| | System issues warning if default surface size is greater than 50% of physical memory because it's a good bet that that amount of memory usage will cause swapping, which is very slow. |
| | Spherical textures didn't use the center point. Now they do. |
| | Added "Journey Though Texture Space". Yep, it's silly, but I like it. |
| 1.13 | Added new options to the "TARGA Superimage" save routine (like fBm noise, ice caps, etc.). |
| | Fixed a minor bug in the BMP surface read that would cause the program to crash sometimes and usually read incorrect data anyway. |
| | Added TIN file type support. |
| | Added PGM file type support. |
| | Added grayscale phase maps and quilt maps. |
| | Fixed pure slope map. Not pretty, but it does more or less what it's supposed to. |
| | Rearranged some of the menus to try to make a little more sense. |
| 1.14 | Added Flood Fill operation for height field data |
| 1.15 | Added simple vector overlays |
| | Added MATLAB MAT File output |
| | Added simple lon/lat ASCII outputs |
| 1.16 | Fixed bug that reversed displayed vertical position from mouse after DTED import. |
| | Added arbitrary quadrilateral sampling. |
| | Fixed some bugs in the 3D triangle/quad output routines. |
| 1.17 | Added arbitrary coloring for height info (defined color sets, etc.). |
| | Made southern ice caps independent of northern ones |
| | New icon! |
| | 256-color ocean now shows depth (shading of data in water) and can have multiple shades if absolute color levels turned off. Comes at the expense of available land colors. |
| | Added Fit Geoid command for the class of programs that assume a flat world but work with real-world data. |
| | Fixed (finally) the problem with the mouse not showing the data under the cursor properly. |
| | Add new options to the mouse position display (now has on/off, decimal lat/lon, DMS lat/lon and pixel position). |
| | Added Line of Sight (LOS) mask calculations. |
| 1.18 | Added arbitrary scaling to the 3D output (DXF, RAW, OBJ, etc.) file formats instead of the generic unit cube style output which didn't work well anyway. |
| | Enhanced Journey Through Texture Space status dialog to actually make sense. Added Keep Parameters dialog to journey when canceled. Also cancels properly now. |
| | When calculating a texture surface, some internal state information wasn't being calculated which caused the Spherical projection dialog to display nothing. |
| | Modified spherical projection dialog to handle fractional resolutions so that the non-inverse mapping mode looked better (when resolution < 1 on low-res maps). And there was a bug caused by hitting ESC on the dialog that's been fixed. |
| | Added 24-bit display mode. Uses the same code base as the TGA Superimage, so the same screen shots can be seen as the data that would have been saved to the file. |

| | |
|---|---|
| | Fixed elevation color computation bug (color was calculated from lowest point on surface, not commanded point or sea level). |
| | Fixed height map gray map option. Now it has the lowest gray value starting at the lowest surface point, not at 0. Leftover bug from long ago (about version 0.1). |
| 1.19 | Added 2D DXF contour export (line segment merging not implemented yet). |
| | Added STM file input/output. This format is the file format required to allow the Scape program to perform mesh reductions on the data sets. |
| | The save surface to grayscale BMP now works directly. |
| | Modified the new lighting property page to get rid of a bunch of parameters that don't make sense in the current configuration. |
| | Added the slope and facing lighting elements as well as some basic blending functionality (numbers must be entered manually, not just easily selectable from the little dialog control). |
| 1.20 | Fixed a problem with single-value scaling (didn't work for scales less than 1). |
| | Added Bryce 2 single-terrain input. Hopefully Bryce 2 output will follow. |
| | Bug in binary import: didn't set the min/max values properly on read. |
| 1.20a | In the 1.20 release, the binary import got badly broken. That means that all the GTOPO30 users won't be able to import data properly. There was also a problem with signed vs. unsigned data during the import. Both fixed. |
| 1.20b | Fixed bug in expression parser that caused expressions of the form "a*b*c" to be interpreted as "a*b" with the "*c" ignored (same with ^ and / operators in groups or the + and - operators in groups). Expressions like "a*b+c*d" or "a*(b*c)" were unaffected. For a recursive-descent parser, that's supposed to be "expr expr" and "term term", not "expr term" and "term factor". Oops. Long-standing. |
| | Fixed spherical projection properly. Now uses only a (correct) inverse transform. |
| | Added large-format spherical rendering. Good for getting maps with a little less distortion. |
| | Spherical projection dialog works properly on 256-color displays. |
| | The Hex and Rectangular grids didn't properly use the degrees/pixels units indicators. Now they do. |
| | The File→Import→Image Subsection didn't properly offset the X destination position. Now it does. |
| 1.21 | Fixed problem with Quad Resample command that inverted pictures. |
| | Added new syntax to fbm command in parser, added dist, besselj, bessely, atan2, and qfrac functions, mod (%) operator. Added remainder of fractal texture types (multi, hetero, hybrid, ridged) to evaluations. |
| | Replaced the Spherical Projection dialog with a Map Projection dialog. Similar in concept, except that a generic reprojection for the texture map is provided, not just an orthographic projection. |
| | Fixed bug that never properly set the final parameter on the multifractal computation type. |
| | Some 7.5 min USGS DEMs didn't import properly due to excess stuff in the header. Should be fixed now. Note that with the 7.5 min DEMs the edge values aren't correct. |
| 1.22 | Added if, eq, ge, le, lt, gt, and, or, xor functions to give a set of conditionals to the program. |
| | Added the xrot and yrot functions to easily enable noise on a perturbed grid (allows vortices). |
| | Added sqr, bias, and gain functions as part of the ongoing miscellaneous function additions. |
| | Fixed surface min/max find bug that required running it several times to get it to complete properly. |
| | Added a rotate 90 degree option. |

| | | |
|---|---|---|
| 1.23 | 5/17/98 | Added a toolbar and cursors that match the current selected tool for the touchy-feely folks out there. Also added pan/zoom and selection tools to the toolbar. |
| | | Added a New option to the File menu. Creates a blank surface with a specified size and edges. |
| | | Added facing and slope operations to expression parser. |
| | | Added selection options and Select menu. Still seem to be some bugs, so don't use this option unless you're willing to have the program behave unexpectedly (and likely crash). |
| | | Added surface cloning brush. |
| | | Fixed misalignment between hex overlay background and foreground. |
| | | Fixed bug that caused advanced form brushes to never draw. |
| | | Added multi-level undo/redo features (on the new Edit menu). Still a few small bugs with repeated undo/redo sequences. |
| | | Added Campaign Cartographer 2 contour map export. |
| 1.24 | 5/27/98 | Fixed undo bugs (scroll position not persistent and GPF on next op after undo). |
| | | Fixed Undo/Redo accelerators. |
| | | Added Abs min value for sea level so that the deepest color can be set to any level. |
| | | Internal reorganization of several systems (no obvious effect on functionality). |
| | | Made adjustments to CC2 contour map export. |
| | | Added crater, scurve, and rmax functions. |
| | | Changed noise semantics (calls correct noise function for 1, 2, 3, or 4 dimensions). |
| | | Fixed bug that wouldn't properly read negative #s from config files. |
| | | Added load/save operations to lighting, brushes, and surface definitions. |
| | | Added (simplistic) Bryce 3D import. |
| 1.25 | 7/12/98 to 11/14/98 | Added painting tools like raise, dig, blur, noise, etc. (actually just access to the main paintbrush tool, but most people never seemed to grasp the purposes of that dialog). |
| | | Added pressure-sensitive tablet support via WinTab library (tested with Wacom ArtZ II and ArtPad II). Still unfixed bug allowing brush Y-resizes with the tablet to protection fault. |
| | | Fixed the problem that did not allow just BMP image maps to be loaded without a surface map. |
| | | Long-standing "feature" that causes a black line at the right and bottom edges of lighted views has been replaced with full computation to the map edges (no more black lines). |
| | | Added status indicators to BMP texture file save and loads. May help with the random garbage bug by performing smaller writes to disk at one time (was just blasting the whole bitmap to disk with one operation). |
| | | Added Bryce 3D file export (that’s the big boost in EXE size). |
| | | Added World Builder Import and reproject. |
| 1.25e | 1/22/00 | Removed all the bits that don’t work right. |
| 1.27 | Late 2004 | Grayscale image surface loads images on disk (jpeg, bmp, gif, etc.) as height fields. |
| | | Color Shift. Interesting texture processing that allows shifting. |
| | | SRTM import including void healer |
| | | Fixed 2D DXF contour export |
| 1.28 | 12/15/04 | Massive restructuring of the code base. Some things may not work as they once did. |
| | | Removed Bryce 2 and Bryce 3 file I/O. 1 MB in the executable for a marginal feature wasn’t worth it. |
| | | Removed 8-bit color support (no more paletted images) |

| | | |
|---|---|---|
| | | Generalized the display code to work better, especially with large images when zoomed in or out. |
| | | Gaussian Blur is now available as a surface operation. |
| | | Adjusted Gray LOS to generate a red overlay on the existing image and a marker at the LOS origin. |
| | | Fixed some memory leaks in the undo code that's been there for years. |
| | | Painting tools have outlines (poorly implemented but it's a first step). |
| | | Set Value point process added to work with selections to form a shape. |
| | | Deterrace area operation to help with maps imported from images. |
| | | Modified Phase map so that it generates angles clockwise from north (black = north, 64=east, 128=south, 192=west). |
| | | Map Projection dialog can be resized for those of you with faster machines. |
| | | Selections now available (painting is slower when selections active). |
| | | Distance selection allows for creating mounds from an arbitrary selection shape. |
| | | Added direction selection (select based on terrain direction). |
| 1.29 | 12/24/04 | Continued restructuring of the code base. Some things may not work as they once did. |
| | | Removed Find Lakes feature. It didn't work as I wanted it so it has moved out for the time being. |
| | | Drainage Areas can be used to find flow patterns on the surface. |
| | | Can convert texture into height field using one of several algorithms. |
| | | Can swap around and translate color channels for textures. |
| | | Read and write 8-bit and 16-bit PNG surfaces; |
| | | Read and write PNG textures. |
| | | Add values to the RGB channels in the texture |
| | | Set values for each of the RGB channels |
| | | Fill Basins in the surface (thanks to Olivier Planchon). 0 gives flat values, tiny positive values give a little slope. |
| 1.30 | 02/03/05 | Map Projection dialog now works again. Some uninitialized variables were causing problems. |
| | | Menus show the icons for toolbars if appropriate (trivial but cute) |
| | | Cleaned up the file open and save dialog to remove some filter errors. |
| | | Correctly connected the save as toolbar button. Oops. |
| 1.31 | 02/05/05 | Fixed 8-way neighbors grayscale texture operation to correctly compute neighbors according to flow rules |
| | | Fixed Drainage Areas texture operation to use correct neighbor computation. |
| | | Added Reverse Drainage Areas texture operation so that rivers can be made to flow uphill. |
| | | Added Compute Basin Deltas area process to figure the difference between the base surface and the basin-filled surface. |
| | | Incise flow into the surface (erosion of a sort). |
| 1.32 | 02/08/05 | Fixed a couple of bugs that were causing crashes and lockups at times. |
| | | Fixed Terragen output to give better results most of the time and added standard output sizes to the Terragen output dialog. |

| | | |
|---|---|---|
| | | Fixed preview slider on the Journey Through Texture Space so that it actually does something. |
| | | Added SVG contour output. Not great quality but it's the same as the CC2 and DXF output except the SVG contours are colored. |
| | | Added limited Binary Terrain (BT format from vterrain.org) reading and writing. It does not support projection information and pads lines to be a multiple of 4 samples wide. |
| | | Fixed Flow Incision to never exceed 100% erosion regardless of the value specified in amount. |
| 1.33 | 02/14/05 | Reorganized the menus. We'll all be confused for a while. The surface processing and calculation routines moved to the Filter menu. |
| | | Added Morphological operators (Erode and Dilate). |
| | | Added RG16 texture operation (copies height of surface into Red and Green 16 bits of the texture). |
| | | Added operation (replace, add, etc.) on the surface calculation dialog. |
| | | Added status bar info on how to use the various mouse-based tools. |
| | | Added mouse-based gradient tool (effect could already be implemented with expression surface, but this makes it easier). |
| | | Added mouse-based floodfill tool (dialog-based one already exists). |
| | | Changed the way that incise flow works to provide generally better control over the process. |
| | | Added Fade to prior operation (allows blending the results of an op with the prior surface to control the strength of the operation). |
| | | Added a Paint Line tool. |
| | | Added unsharp mask filter. |
| | | Fixed a GPF error on loading a PNG surface that wasn't square |
| | | Removed the limitation that required surfaces be a multiple of 4 wide (it was caused by the old 8bpp textures used early on). |
| | | Binary File open is back. Seems I took it out by accident a while back. |
| 1.34 | 02/21/05 | Added min and max operations for tools that have mathematical blending operators (except brushes) |
| | | Added new gray maps types for gradient, c0, and c1 discontinuities from Saito & Takahashi's 1990 paper "Comprehensible Rendering of 3D Shapes" |
| | | Image brush is working now (loads an image as a grayscale brush) |
| | | Filter>>Other>>Apply image applies an image from disk to the current surface. |
| | | Removed flood fill as an option on the filters section. you have to use the mouse-based tool now, sorry. |
| | | Added a toolbar with options for the basic drawing tools. It still has some minor problems, though (titles on the toolbar, for example). |
| | | Added Filter>>Other>>Place Brushes in Pattern. Can lead to amusing results, especially with image brushes. |
| | | Removed the apply option from the brushes. It wasn't really useful with the advent of selections a few versions back. |
| | | PNG files were being saved inverted vertically. Now they save correctly. |
| | | Added autorotate option to painting brushes. Only works for the brush type, not the noise or convolve types. It can have some odd undersampling effects. |
| | | Added simplistic preset manager to brush dialog and toolbar. |

| | | |
|---|---|---|
| 1.35 | 02/28/05 | Added border option to selection. Doesn't always get the exact specified border width, but it's usually close. |
| | | Added binarize option to selection. Converts feathered selection to hard-edged selection. |
| | | Added single row selection tool. |
| | | Added single column selection tool. |
| | | Added polygon selection tool. |
| | | Dropdown toolbar items are now available for some of the toolbar items (click and hold on a button for a second to see). |
| | | Added toolbar options for selections (selections can auto-feather now and ellipse and rectangle have aspect or fixed size). |
| | | Blur and Sharpen tools now use the last undo level as the source. This change gives more control over the results, but can be more tedious to use. |
| | | Clone tool is a little cleaner internally. You may notice a small change in the way it works. |
| | | Added selection render hex grid and rectangular grid. |
| | | Added move selection tool. Keyboard move operations clip too early at the edge of the display. It's slower than it needs to be. |
| | | Added magic wand selection tool. |
| 1.36 | 03/07/05 | Removed Altitude Key Dialog (wasn't working properly). |
| | | Now properly flushes the undo stack on file load and new file. |
| | | Fixed painting toolbar on/off command. Also shows/hides tool options bar along with the main toolbar (it was possible to lose the options bar by undocking and closing it). |
| | | Terragen export works better: output values almost correct, correct size of file saved (not always 513x513), and allows curved output. |
| | | Fixed brush and selection outlines when zoomed and scrolled. |
| | | Fixed flood fill tool. Now faster, less likely to fail, and has no limitations on the low, high, and new values. |
| | | Fixed the clone tool to have a toolbar and show brush outlines (behaves like a regular brush, almost). |
| | | Added crop to selection tool. |
| | | Moved surface tools to their own submenu to free up some shortcut keys. |
| | | Reorganized some dialogs to make room for help buttons. Some of them still need some layout work. HTML-based online help will be included in a version real soon now. |
| | | The Texture option on the light blending page will now accept a standard image file as well as a large RGB raster block. |
| | | Added operation to multi-file import to allow import and processing at the same time. |
| | | Added output directory option to sea flood operation. |
| 1.37 | 03/08/05 | Fixed output bug so that save as works correctly on all machines. |
| 1.38 | 03/14/05 | Added first cut at help system, HTML files stored in the help directory. The code's there, I just haven't done the help file parts yet. Any volunteers? |
| | | Added mound tool. |
| | | Fixed problem with texture image in lighting model (wouldn't load more than one image in a session). |
| | | Added a preference to show the mouse info at the mouse position as well as the status bar. The flicker is annoying, though. |
| | | Reworked the blending page of the lighting model to make it a little clearer, I hope. At a |

| | | |
|---|---|---|
| | | request from a user, F5 will toggle the color map on and off and F6 will toggle the texture image on and off for lighting. Also picks up the edges of the current map on the first texture specification to make things easier. |
| 1.39 | 03/21/05 | Fixed height problem with mound when doing non-linear operation. |
| | | Added Preview to height field computation. |
| | | Added Fractal Parameters option to mound tool. |
| | | Added arbitrary profile on mound tool. |
| | | Fixed Cancel on mound tool. |
| | | Fixed menu display on Undo, Redo, and Fade to Prior. |
| | | Fixed undo messages for Freehand and Polygon Selections. |
| | | Added altitude remap tool (Filter >> Other >> Remap Altitudes) |

# 10. Known Bugs

The following bugs (they're documented, so they're actually "features") are known. If there is a workaround, it will be given below.

- The expression parser can't handle certain types of expressions and crashes the program sometimes. "cos(r*r*" is a known example.
- Text output routines don't properly save all of the data. They output only 0 through 1 step less than the surface dimension in X and Y. (0 to width-width/(new width), 0 to height-height/(new height)).
- The display code for the orthographic projection is upside down. As a result, the top of areas in the northern hemisphere are wider than the bottom, when they should be narrower. The easy fix (that the user can do while working with it) is to swap the top and bottom edge values, then change the sign. For example, if the top is 40 degrees and the bottom is 30 degrees, change the top to -30 degrees and the bottom to -40 degrees.