

***Barebones RIMS Troubleshooting Manual v1.0***  
***Oct. 30, 2006***

***Copyright (C) 2006, Eric Cowgill***  
***RIMS program, Copyright (C) 2006, Tony Bernardin***

## **General Notes**

\*Take notes as you process images, attempt to quadbuild, and run RIMS. Things will mess up and it will be helpful if you can look back and determine what you did that did (or didn't) work several months, weeks, days, or hours before.

\*If a process or procedure fails, try it again to make sure the problem is reproducible. Make sure you keep clear notes detailing the steps you took on procedures that proved successful in the past.

## **Quadbuild**

### **QB takes a long time!**

\*For big files, this is ok. For example, a 750 MB bil texture takes 1125 s (19 min) to run on an AMD Athlon 64X2 dual core, 2.4 GHz machine with 2.96 GB Ram running WinXP.

### **QB fails during reading header**

\*There may be a software incompatibility with this particular machine. First, try running Quadbuild using the example \*.bil files distributed with RIMS. If that fails, it's your machine. If that runs, it's something with the input files, in which case you should check them to make sure you've satisfied all the criteria listed in the Quadbuild Manual.

\*This error will be generated if your Texture file is only one band, instead of the 3 band that Quadbuild expects. Note that even grayscale images must be in 3 band format (i.e., R = G = B).

### **QB fails during reading image**

\*Try making and running a very small (few Mb) test bil to make sure your data and processing procedure will work.

\*Make sure data are in correct format (byte vs. unsigned integer).

\*This often happens when file size is too big. Try reducing the quad resolution (i.e., change from "ortho 128" to "ortho 64" in the input).

\*The max .bil file size is 950 Mb, do not exceed this, it will waste your time and lead to high levels of frustration. Although large RIMS projects do provide an impressive ability

to zoom from overview to highly detailed views, the visualization does become a bit quirky, so it's generally better to strive for smaller RIMS project files.

### **QB fails during building Quads**

\*try dropping the quad resolution (e.g., from 128 to 64 or from 64 to 32). On the AMD Athlon machine described above, a 750 Mb texture bil failed to run at "ortho64" but then worked at "ortho 32".

## **RIMS**

If RIMS launches but the display is clearly messed up (scrambled tiles, black screen, etc.), it's usually a random blunder in the data prep. First, quadbuild the example bil files distributed with RIMS to make sure the distribution version of RIMS is working. Then start from scratch and regenerate a new set of bil input files. This often works and is generally faster than trying to track down the error in the processing of the previous set. However, if the error keeps appearing, then it's likely a systematic error with the way the bil files are being prepared, which will require systematic experimentation to track down the problem with your data preparation protocol. In this case, it's best to start with a RIMS project that just uses a dem and a resampled hill shade image that was made from that same dem, because then the image dimensions will match.

### **Terrain view flashes open and then closes upon launch**

\*This can be a problem with the OpenGL configuration on some systems. The following OpenGL Extensions must be supported by your graphics system:

- GL\_ARB\_vertex\_program
- GL\_ARB\_vertex\_buffer\_object

### **Terrain display window opens but is black (cursor is tiny or missing)**

\* First, try zooming in on the data to see if it appears.

\*The most common cause of this is that the input files had lat/long values in units of degrees but elevations in meters. Check the files used to make the bils for Quadbuild to see if this is the case.

\*This can also happen if the texture is bad. Try remaking this, and then try redoing the DEM

\*This may also happen when the bil file was not saved in the correct format (byte vs. unsigned integer)

### **Texture is scrambled**

\*This can happen when the texture size (in terms of the number of rows and columns) is not twice that of the dem. If the DEM is 4000 rows by 3000 columns with 30 m pixels, then the texture should have 8000 rows by 6000 columns with 15 m pixels. If not, then the dem and texture quadtree levels will be mismatched and the display is scrambled.

While the rows and columns do not need to differ by *exactly* a factor of two, they do need to be close (within a few pixels).

\* This can also happen when round off errors result in a discrepancy in the number of quad levels between the DEM and texture, so that RIMS ends up matching the wrong texture quad level with a given DEM quad level in the display (see the text of the G-Cubed paper for a explanation of quadtrees). It often works to reprocess the DEM and texture data using slightly larger or smaller dimensions (either in terms of pixels or geographic area) for the project area and then regenerate the \*.bil input files and re-run the Quadbuild program. By tweaking the data dimensions in this way you change the number of rows and columns of DEM and texture data and hopefully get to the point where the texture and DEM quadtrees are matched.

\*The best thing to do is to crop one of the files during preprocessing to the spatial dimensions of the other using the crop to file function in ENVI (although there can still be small discrepancies in the number of rows and columns of data between the DEM and texture data sets that can mess up quadbuild):

- main ENVI menu > basic tools > resize data (spatial/spectral)
- select file to crop and then “spatial subset”
- subset using > file
- select the file with dimension that cropped file should match

\*This may also happen when the \*.bil file was not saved in the correct format (byte vs. unsigned integer)

## **Exporting Mapping**

### **e00 file import into Arc stalls**

\* This can happen when one of the mapped lines has more than 400 nodes. To check if this might be a problem, open the e00 file in a text editor and check the entries to see if any of the lines has too many nodes. If so, then either delete the line inside RIMS, split the line by manually editing the e00 file in the text editor, or try to import it into Arc via another route.

\* This can also happen when your base projection is unusual and Arc has a hard time coping. Our experience has been that Arc can handle most user-defined, run-of-the-mill variants of a Transverse Mercator projection (except those with negative x, y, or z values). However, we have failed to get it to import e00 files mapped in skewed projections.

### **ArcToolbox e00 import never ends**

\* The best way to get e00 files into Arc is to use ArcCatalog: Launch ArcCatalog, Select View>Toolbars and make sure ArcView8x tools are turned on. This gives you a "conversion tools" pull down menu. One of the options on this menu is “Import from interchange file”. This seems much better than the toolbox function. You can also run it in batch mode.

\* When the ArcToolbox version of the e00 import utility stalls it leaves the process running on the CPU, even if Arc is forced to quit. Each e00 import takes 50% of the CPU, so when it tanks you can try one more time but if it tanks again you have to either kill the process or force a reboot of the system because the CPU is pegged at 100%. Pressing ctrl+alt+del opens the windows task manager and switching to the performance tab shows the CPU status. If CPU is pegged, the best choice is to select the “applications” tab and then kill the Arc application. An alternative is to select the “processes” tab and then find and kill the hanging process there. The “nuclear” option is to simply reboot the machine.

\* Last ditch attempt to get your mapping out: If all else fails and you really want your mapping out, try using ENVI to produce a vector file that is readable by ARC.

- open ENVI, select Vector > Open Vector File
- pick “e00 file” from the file type and find the problematic e00 file and open it.
- save this imported file as an ENVI vector file (EVF).
- load this ENVI vector file into a viewer
- select Vector > Convert EVF to DXF. DXF is a Cad vector file that it seems ARC can read regardless of what strange projection it may be in. Note: All attributing will be lost in this process, but the lines will be exported in their correct projection and position.