HYPACK
a xylem brand

Sounding Better!

New HYPACK® File Formats in the 2018 Release

By Bill Bergmann

Depending on the type of data files you are using in your project, you might notice some new file formats on your computer and be curious about them. This document will explain what they are and the reason for their existence.

IMG

The shell can display a variety of raster file types and each type can come in a variety of encodings — number of bits per pixel, color depth, compression type etc. This led to a large amount of internal code to handle all the permutations. To simplify matters, increase speed and reduce resources, the shell now converts all raster types to a common "image" format with an extension of IMG. Specifically, we tack the IMG extension onto the end of the filename of the original raster format. For instance 'SomeFile.png' will produce the image file 'SomeFile.pngIMG'.

So what are the benefits besides simplified code inside of HYPACK®? I mentioned speed and resources. The image format is in a form designed for rapid display so nearly no additional processing is required to get it on the screen. It has a rapid, randomly accessible nature such that it doesn't need to remain in memory, saving a significant amount of resources. Whether you have 1 or 100 raster images active, the memory required is the same, since the image is accessed just in time, not permanently!

A nice side benefit is a common set of manipulation options for all raster data. For instance, Transparency and Export to Google Earth are available for any supported raster format. Currently, the IMG form of a raster file is a permanent part of your project so, once it exists, you should notice projects using raster data load much faster. Also, toggling the enabled state of such data is very quick.

TEX

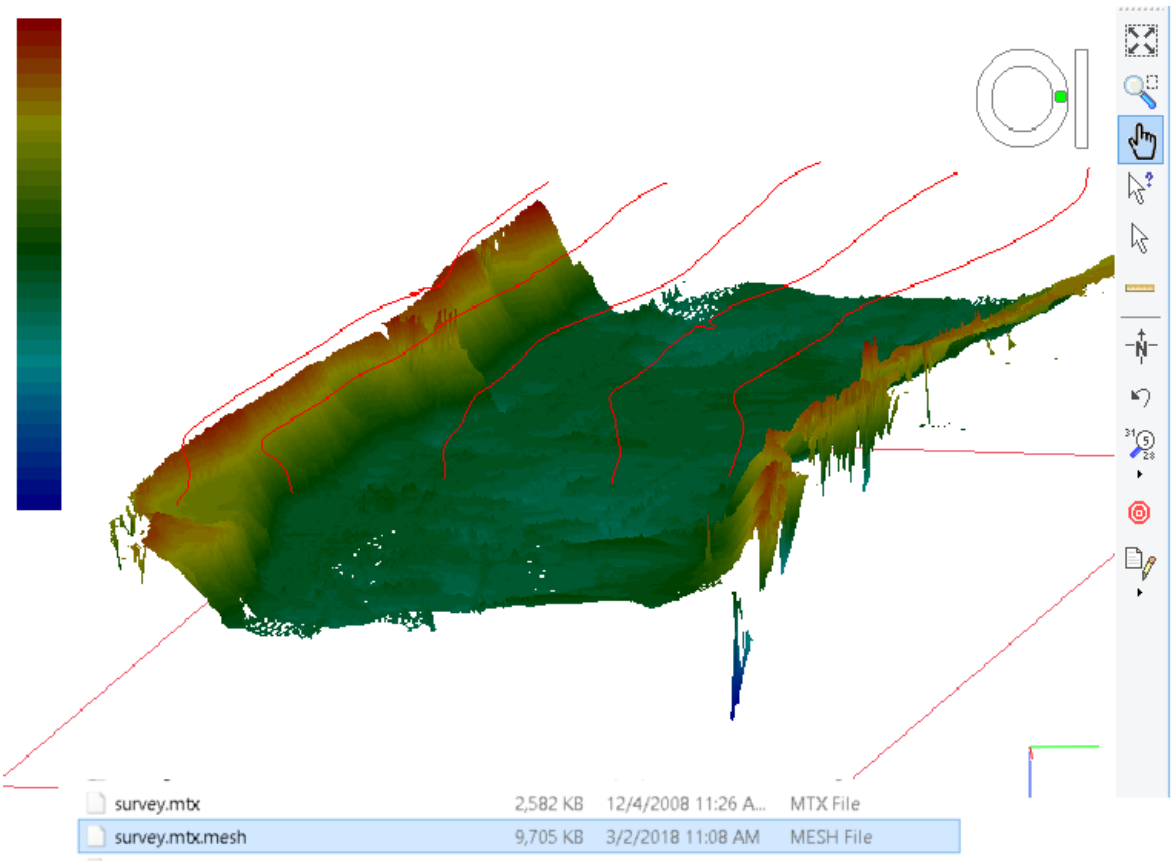
This is a further refinement of the IMG format used only for 3D display mode. Here's why we need it:

In 3D mode, HYPACK® uses the OpenGL library for drawing. Displaying raster data in OpenGL requires the use of a construct known as a Texture. The size of a Texture is limited by your computer hardware—typically 1 megabyte on a standard setup, and up to 8 or so megabytes on high end computers in hardware accelerated mode. In order to get images that exceed the available texture size into OpenGL we need to split the image into a series of small enough images (think tiling). This is what the 'texture' format is. It is the 'image' format tiled enough to fit inside your computer's Texture limitation.

All the benefits listed above for IMG files apply to TEX files. Currently, in an effort to reduce hard drive space used, the TEX file exists only for enabled files and only while HYPACK® is running. This isn't too bad as the conversion from IMG to TEX is very quick. The TEX file is stored in your HYPACK® install Temp folder.

MESH

FIGURE 1. *Sample MESH File*



MESH files are currently used only for matrix files in 3D display mode. Figure 1 shows a matrix displayed in 3D mode and the file naming relationship in your project folder. The algorithm for displaying a 3D model of a matrix is to treat the center of the cell as a fixed depth position. We then interpolate a depth at the 4 cell corners using neighboring cells. All the 3D points, cell center and corners are then connected up in triangles for display by OpenGL. This methodology produces a nice transitional smoothing effect while honoring the cell depths.

To avoid recalculating this information at every draw cycle, the conversion is done once and stored in a rapid load MESH file. Like the IMG and TEX formats, we tack the extension onto the existing filename. Using this format, we again save memory as the entire matrix is no longer kept in memory, but rather accessed just in time as required. This format, like the TEX format, is only kept around while HYPACK® is running.