Optimizing Textures

> In many cases, users no longer need to use **tdlmake** to manually optimize their textures as 3Delight now does it automatically. This is explained in details in Handling of Textures.

tdlmake preprocesses TIFF, PNG, JPEG, GIF, IFF(6), SGI, PIC(7), PSD(8), TGA(9), "bake," Radiance and OpenEXR files and convert them into an efficient texture format suitable to 3Delight for rendering. It can also convert a zfile into a shadow map. Optimizing textures using **tdlmake** is important for these reasons:

1. **tdlmake** creates a mipmapped version of the original texture, allowing *3Delight* to produce nicer images more efficiently.
2. *3Delight* employs a caching mechanism for texture data which works well with tiled images. Using raw (striped) non-converted TIFFs may degrade overall performance.

> The optimized texture is saved into a regular TIFF file format that can be viewed with any image viewer. For clarity, we recommend using a **.tdl** extension for 3Delight texture files.

Command Line Options

tdlmake is invoked by specifying at least two file names and an optional set of command-line switches:

```
% tdlmake [options] input.tif [input2.tif ... input6.tif] output.tif
```

Option	Description
Output Format	
`-envlatl`	
`-envcube`	
`-twofish`	
`-lightprobe`	
`-dirtex`	
`-shadow`	
Color Profiles	
`-colorspace space`	
`-gamme g`	
`-rgbagamma r g b a`	
Compression	
`-lzw`	
`-deflate`	
`-packbits`	
`-c-`	
Texture Filtering	
`-filter f`	
`-filterwidth n`	
`-sfilterwidth n`	
`-tfilterwidth n`	

-window w	
-blur n	
-scale n	
-qality	
-nomipmap	
Texture Wrapping Modes	
-smode	
-tmode	
-mode	
Output Data Type	
-byte / -sbyte	
-short / -sshort	
-float	
Image Orientation	
-flipst	
-flips	
-flipt	

Working With Large Textures

tdlmake has been designed to work with any textures, including very large ones. One exception is compressed TIFFs that have a large "rows per strip" value. Here is an example output of **tiffinfo** on a large texture file that can cause problems for **tdlmake**:

```
% tiffinfo earth.tif
  Image Width: 43200 Image Length: 21600
  Resolution: 72, 72 (unitless)
  Bits/Sample: 8
  Compression Scheme: AdobeDeflate
  Photometric Interpretation: RGB color
  FillOrder: msb-to-lsb
  Software: "ImageMagick 5.5.7 07/22/03 Q16 http://www.imagemagick.org"
  Document Name: "earth.tif"
  Orientation: row 0 top, col 0 lhs
  Samples/Pixel: 3
  Rows/Strip: 21600
  Planar Configuration: single image plane
  Predictor: horizontal differencing 2 (0x2)
```

There is only one strip, made of 21600 (total image height) rows. This means that accessing any scanline in this TIFF forces the TIFF reading library to uncompress the entire file(12). In order to lower memory usage for such large files, it is suggested that a lower "rows per strip" count be used. Typically, 16 or 32 scanlines is a good choice. **tdlmake** prints a warning if it encounters a file that has the aforementioned problem:

```
tdlmake: warning, reading texture 'earth.tif' may take a large
tdlmake: amount of memory. Please refer to user's manual if you are
tdlmake: unable to convert this file.
```

Examples

Here are some examples using **`tdlmake`** on the command line:

To create a texture named `grid.tdl` from a TIFF named `grid.tif` using a gaussian downsampling filter of width 4:

```
tdlmake -filter gaussian -filterwidth 4 grid.tif grid.tdl
```

To create a cubic environment map in which all cube sides were rendered using 90 degrees field of view:

```
% tdlmake -fov 90 -envcube \
  in1.tif in2.tif in3.tif in4.tif in5.tif in6.tif \
  envmap.tdl
```

or (won't work in a DOS shell):

```
tdlmake -fov 90 -envcube in?.tif envmap.tdl
```

To create a texture using the high quality downsampling mode and show progress while doing so:

```
tdlmake -progress -quality high grid.tif grid.tdl
```

To create a shadow map from a zfile (The zfile display driver):

```
 tdlmake -shadow data.z shadowmap.tdl
```