

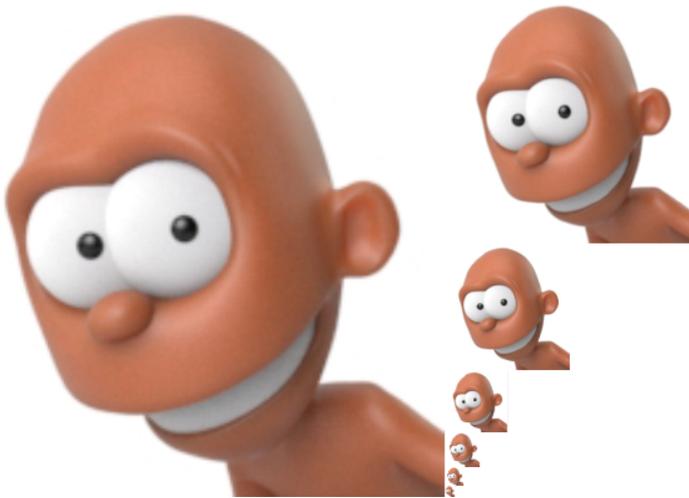
## Maketx

Maketx is a command-line utility to convert images to tiled, MIP-mapped textures, similar to txmake in Pixar Renderman. It is part of OpenImageIO (<http://www.openimageio.org>) and was developed by Larry Gritz at Sony Pictures Imageworks.

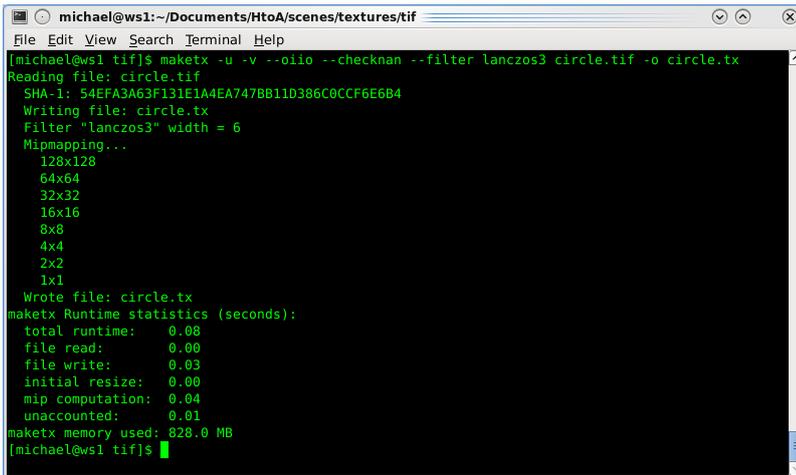
### Benefits of Using .TX

The first benefit is that you are assured of having mipmaps and tiles. These will dramatically improve time to the first pixel, overall render time, and allow using a smaller texture cache. This should be considered mandatory, with the possible exception being don't use .tx for the few images you are actively modifying if you don't want to wait for them to be constantly converted to .tx every time you make a change.

The previous level you could get from non-tx files that you saved with tiles and mipmapping. The second level of benefit is only with .tx files and that involves maketx adding metadata that lets Arnold make more optimizations, such as being able to detect duplicate textures and only loading a single copy into memory or detecting constant color images, such as an all-black **UDIM**, and instead of storing all those black pixels in memory Arnold can special case that.



An example of mipmap image storage. The first image on the left is shown with filtered copies reduced in size.



```
michael@ws1:~/Documents/HtoA/scenes/textures/tif
File Edit View Search Terminal Help
[michael@ws1 tif]$ maketx -u -v --oiio --checknan --filter lanczos3 circle.tif -o circle.tx
Reading file: circle.tif
SHA-1: 54EFA3A63F131E1A4EA747BB11D386C0CCF6E6B4
Writing file: circle.tx
Filter "lanczos3" width = 6
Mipmapping...
 128x128
 64x64
 32x32
 16x16
 8x8
 4x4
 2x2
 1x1
Wrote file: circle.tx
maketx Runtime statistics (seconds):
total runtime: 0.08
file read: 0.00
file write: 0.03
initial resize: 0.00
mip computation: 0.04
unaccounted: 0.01
maketx memory used: 828.0 MB
[michael@ws1 tif]$
```

The maketx command with verbosity, converting from a *tif* to a *tx*

A tutorial that shows the workflow involved when working with *.tx* files can be found [here](#).

## Usage

`maketx [options] file...`

eg: `maketx -v -u --oiio --checknan --filter lanczos3 path/to/fileIn.tif -o path/to/fileOut.tx`

When pre-processing your texture files with **maketx** make sure you add the **--oiio** flag which will generate TX files optimized for Arnold that can render even faster by forcing the tile size to be optimal for OpenImageIO (64x64). It also stores channels interleaved (RGBARGBA...) making them faster to read and enables constant color detection: if the image is a constant color it will convert it to a tiny single color *.tx* file. Additional metadata and an image hash (fingerprint) are embedded by default. The fingerprint lets OIIO detect duplicate images (not tiles) when opening new files. Generally using maketx will result in better texture I/O performance than using txmake.

Note also that if you need your textures to stay compatible with Renderman, there is a **--prman** switch that will ensure the tile size matches what Renderman expects.

If the output color space is not the same bit depth as input color space, then the data format needs to be set to the proper bit depth using the **-d** option.

The best workflow is to always generate *.tx* files using maketx as the last step before rendering and

use those .tx files just for rendering. This will ensure the best performance. Also, ensure that your textures are in linear color space before converting them with maketx. No information is lost when converting High Dynamic Range images to the .tx format.

## Parameters

| flag  | description  |
|---|--|
| --help  | Print help message   |
| -v  | Verbose status messages  |
| -o %s   | Output filename  |
| --threads %d  | Number of threads (default: #cores)  |
| -u  | Update mode  |
| --format %s   | Specify output file format (default: guess from extension)   |
| --nchannels %d  | Specify the number of output image channels.   |
| -d %s   | Set the output data format to one of: uint8, sint8, uint16, sint16, half, float  |
| --tile %d %d  | Specify tile size  |
| --separate  | Use planarconfig separate (default: contiguous)  |
| --fov %f  | Field of view for envcube/shadcube/twofish   |
| --fovcot %f   | Override the frame aspect ratio. Default is width/height.  |
| --wrap %s   | Specify wrap mode (black, clamp, periodic, mirror)   |
| --swrap %s  | Specify s wrap mode separately   |
| --twrap %s  | Specify t wrap mode separately   |
| --resize  | Resize textures to power of 2 (default: no)  |
| --noresize  | Do not resize textures to power of 2 (deprecated)  |
| --filter %s   | Select filter for resizing (choices: box triangle gaussian catrom blackman-harris sinc lanczos3 radial-lanczos3 mitchell bspline, disk, default=box) |
| --nomipmap  | Do not make multiple MIP-map levels  |
| --checknan  | Check for NaN/Inf values (abort if found).   |
| --Mcamera %f %f %f %f<br>%f %f %f %f %f %f %f %f<br>%f %f %f %f | Set the camera matrix  |
| --Mscreen %f %f %f %f<br>%f %f %f %f %f %f %f %f<br>%f %f %f %f | Set the camera matrix  |
| --hash  | Embed SHA-1 hash of pixels in the header (deprecated. hashes are always computed).   |
| --prman-metadata  | Add prman specific metadata  |
| --constant-color-detect   | Create 1-tile textures from constant color inputs  |
| --monochrome-detect   | Create 1-channel textures from monochrome inputs   |
| --opaque-detect   | Drop alpha channel that is always 1.0  |
| --stats   | Print runtime statistics   |
| --mipimage %s   | Specify an individual MIP level  |
| <b>Basic modes (default is plain texture)</b>                   |  |
| --shadow  | Create shadow map  |
| --envlatl   | Create lat/long environment map  |

|                                   |  |
|-----------------------------------|--|
| <code>--envcube</code>            | Create cubic env map (file order: px, nx, py, ny, pz, nz) (UNIMP)  |
| <b>Color Management Options</b>   |  |
| <code>--colorconvert %s %s</code> | Apply a color space conversion to the image. If the output color space is not the same bit depth as input color space, it is your responsibility to set the data format to the proper bit depth using the -d option. (choices: linear, sRGB, Rec709) |
| <code>--unpremult</code>          | Unpremultiply before color conversion, then premultiply after the color conversion. You'll probably want to use this flag if your image contains an alpha channel.   |
| <b>Configuration Presets</b>      |  |
| <code>--oiio</code>               | Use OIIO-optimized settings for tile size, planarconfig, metadata, and constant-color optimizations.   |
| <code>--prman</code>              | Use PRMan-safe settings for tile size, planarconfig, and metadata.   |

Do not linearize textures used to drive scalar values as you will lose precision values.

The maketx utility is available in the MAXtoA plugin folder.

In Windows this would be:

```
C:\solidangle\maxtoadeploy\2015\bin\maketx.exe
```