

Preparing Textures for Rendering

Texture File Format

For *3Delight* to render efficiently with texture maps, texture file must be prepared and saved in a certain format called "TDL", this format is a standard TIFF file but it is written in *tiles* rather than *scanlines*, it is augmented with "mipmaps" and finally it can take into account a *color profile* transformation.

It is also possible to prepare textures in the *tiled*, *mip-mapped*, EXR format.

This preparation can be either automatic or done manually.



Although the *3Delight* texture format is standard TIFF augmented with mipmaps, some applications are not able to read it properly because the data is stored in *tiles* and not *scanlines* (this is a performance feature). Refer to the `-preview` and `-preview8` options of `tdlmake` command used below in this document which allows for such textures to be read in applications that cannot read *tiles*.

Automatic Preparation through the Hypershade shading network

If an object is shaded using an *Hypershade* shading network that uses a texture file (any image file format via a node such as the *File 2D texture* node), *3Delight* will automatically convert the texture to *3Delight*'s efficient texture file format and then use this file for rendering.

The automatic conversion is skipped if the file is already prepared. It is detected as already prepared in the following three situations:

1. The file extension is: `.tdl`
2. The file extension is: `.tdl.tif`
3. The file extension is: `.tdl.exr`

All the textures that are prepared using this automated process are cached in the location specified in [3Delight Data Locations](#). For subsequent renderings (of the same image, all the frames of an animation or even for other scenes or projects) *3Delight* will re-use the prepared textures from the cache location. If any original texture is modified, it will be converted again to update the cached version. All this process is automated.

The automatic conversion will automatically linearize file textures according to whether input color textures are specified as non-linear in the color management settings of Maya. In such case, *3Delight for Maya* will automatically linearize all Maya File textures which inherit the global color profile settings. In a (rather typical) scenario where input color textures are coming in Maya both as non linear (color data from PhotoShop) and as linear (non-color data from ZBrush/Mari/Mudbox and IBL HDRI data), *3Delight for Maya* will automatically skip linearization for what is supposed to be linear

- displacement
- bump

Special cases where the user should manually set as linear color profile in the Maya file texture are:

- HDRI maps fed to `delightEnvironmentShape` nodes
- stacked bump nodes (wether set as bump maps or normal maps)
- anisotropy maps

It is possible to prepare the texture file manually (and save it using a file name with the above extension) in order to avoid this automatic conversion – which could slow down the rendering process for when textures are not already converted. This procedure is explained in the following section.

Manual Preparation of Texture Files

3Delight comes with a command line tool named `tdlmake` to prepare texture file for efficient rendering. The prepared texture file should be saved using a file name with the `.tdl` extension. You can also use the `.tdl.tif` extension instead if you want to keep the ability to open and view the texture using the tool of your choice (if it supports the TIFF image format with data stored in *tiles*). Additionally one can also prepare textures in the *tiled*, *mip-mapped*, EXR format using the OpenImageIO (OIIO) command line program `oiiootool` which provide finer-grained controls for color conversion using OpenColorIO (OCIO).



`oiiootool` is not distributed with *3Delight*, refer to the [OpenImageIO](#) website and repository for more informations.

The prepared texture file can then be used directly in a *Hypershade* node such as the *File 2D texture* node.

Here is a typical example of how to use the `tdlmake` command line tool:

```
tdlmake -preview8 0.1 mytexture.tif mytexture.tdl.tif
```

Here is a typical example of how to use the `oiiootool` command line tool, note that the wrap mode periodic is required by *3Delight*:

```
oiiootool mytexture.exr --tile 16 16 -otex:wrap=periodic -o mytexture.tdl.exr
```

When manually preparing your "mipmap" texture files, you should consider to linearize your non-linear input textures:

- For [non-linear color maps](#) which are typically coming from Adobe PhotoShop (or other non-linear applications) as TIF files in the sRGB color space:

```
tdlmake -colorspace sRGB -preview8 0.1 mytexture.tif mytexture.tdl.tif
```

```
ooiotool mytexture.exr -colorconvert sRGB linear --tile 16 16 -otex:wrap=periodic -o mytexture.tdl.exr
```

or REC/BT.709 color space:

```
tdlmake -colorspace BT.709 -preview8 0.1 mytexture.tif mytexture.tdl.tif
```

```
ooiotool mytexture.exr -colorconvert Rec709 linear --tile 16 16 -otex:wrap=periodic -o mytexture.tdl.exr
```

- For [non-linear color maps](#) using the OCIO large variety of named color profiles, refer to the [OpenColorIO](#) website, we provide a simple example as a reference here:

```
ooiotool mytexture.dpx -colorconvert lg10 lmf --tile 16 16 -otex:wrap=periodic -o mytexture.tdl.exr
```

- For [linear non-color maps](#) which are typically coming from Mari/Zbrush/Mudbox as EXR files (therefore already linear) there is no need to perform any colorspace conversion since by default data is interpreted linearly by the texture optimizations tools. You should therefore keep the following textures in linear color space:

- float/vector non-color displacement maps
- non-color bump maps
- non-color normal maps
- non-color anisotropy maps
- non-color masks
- non-color baked non-color maps

For reference these commands will explicitly leave the data in linear format:

```
tdlmake -colorspace linear -preview8 0.1 mytexture.tif mytexture.tdl.tif
```

```
ooiotool mytexture.exr -colorconvert linear linear --tile 16 16 -otex:wrap=periodic -o mytexture.tdl.exr
```

Refer to section "3.3 Using the Texture Optimizer - tdlmake" in the [3Delight-UserManual.pdf](#) and to [openimageio.pdf](#) for more details on these command line tool by running the following command in your terminal/prompt:

```
tdlmake -h
```

```
ooiotool --help
```

For the tdlmake case, this process can be automated furtherly via MEL using the following function in *3Delight for Maya* 2015/2016 (or newer):

```
DL_convertTextures()
```

This function will scan all the shaders and then do the necessary work to call tdlmake.

For a proper *linear workflow*, color textures must always be linearized before rendering, this is a complex topic and depends on Maya versions, see:

- [color management in 3Delight for Maya 2015 and older](#)
- [color management in 3Delight for Maya 2015 Extension and newer](#)



Important

Maya can not read TIFF file stored in *tiles*, only in *scanlines*. For this reason, it is necessary to use tdlmake with the '-preview' or '-preview8' options so it will also insert an embedded thumbnail (stored in *scanlines*) at the beginning of the file that Maya can read. This is useful for Maya to display a texture preview image.

