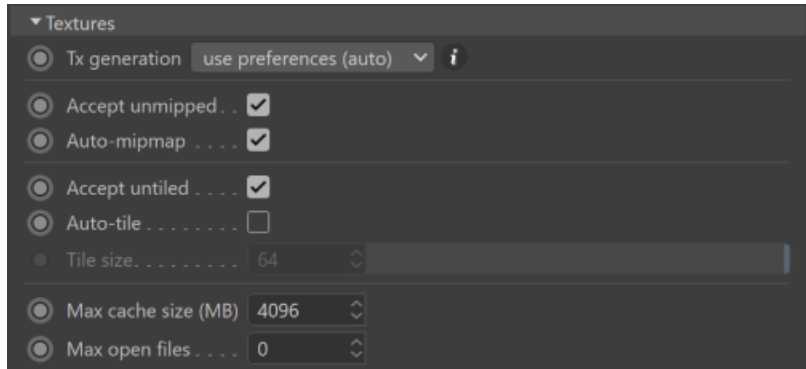


Textures (settings)

The Textures group of controls lets you specify how to handle texture files in Arnold.

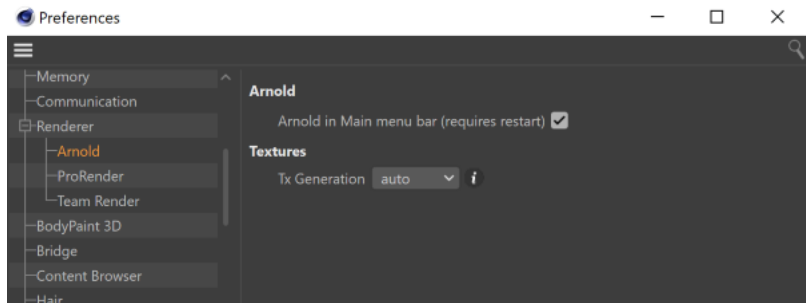


Tx Generation

Tx is a tiled and mipmapped texture format, which will dramatically improve time to the first pixel, overall render time, and memory usage. Therefore converting textures to Tx files is highly recommended.

You can choose your preferred workflow of generating Tx textures:

- **use preferences:** Use the settings defined in the global [Preferences](#).



- **none:** No Tx files are generated, the original textures are used. Note that this mode is recommended in special cases only because it can hurt performance.
- **manual:** Use the [Tx Manager](#) to generate Tx files manually when needed (e.g. original texture has been modified, color space has been changed, etc.). Existing Tx files are used in the render instead of the original images.
- **auto:** Textures are automatically converted to Tx files before rendering the scene. Textures are converted from the [image color space](#) to the [render color space](#). Each 8-bit texture is converted to 16-bit float data type images to avoid quantization errors in dark tones. This are the options passed to the maketx tool:

```
-u --unpremult --oioo (--colorconvert [image_space] [render_space] --colorconfig [config]) (--format exr -d half --compression dwa)
```

Accept Unmipped

High-resolution unmipped texture maps are very inefficient to render because the highest resolution level must be loaded into memory regardless of the distance rather than a lower resolution level. When this flag is disabled, any attempt at loading an unmipped file will produce an error and abort the renderer.

Auto mipmap

If a texture map file is not natively mipmapped (e.g. a JPEG file), enabling this option will trigger the generation of a mipmap on demand. The output will be stored in memory and put in the global texture cache. This mipmap generation increases render time, especially for scenes with many high-resolution textures. To avoid this performance slowdown, it is recommended to use texture file formats that natively support mipmapping (such as TIFF and EXR), where the mipmaps are preprocessed before the render and stored in the file. This preprocessing can be done with the [maketx](#) utility.



Disabling 'options.texture_automip' will result in unfiltered non-mipmapped textures.

Accept Untiled

If a texture map file is not natively mip-mapped, the render will give an error unless this option is checked.

Auto-tile

If a texture map file is stored in scanline mode (e.g. a JPEG file), enabling this option will trigger the generation of tiles on demand. The output will be stored in memory and put in the global texture cache. This process increases render time, especially for scenes with many high-resolution textures. To avoid this performance slowdown, it is recommended to use texture file formats that natively support tile mode (such as TIFF and EXR). You can create tile textures with the [maketx](#) utility.

Tile Size

This is the size of the tiles when using auto-tiling. Bigger values mean less frequent texture loads but use more memory.

Max Cache Size (MB)

The maximum amount of memory to be used for texture caching.

Max Open Files

The maximum number of files that the texture system keeps open at any given time to avoid excessive closing and re-opening of files when caching individual texture tiles. Increasing this number may result in slightly better texture caching performance. On the other hand, if this value is higher than the maximum number of concurrently open files supported by the operating system (e.g. Linux vs Windows), some texture lookups may fail and return red. This is set by default to 0, which means that the maximum number of texture files that can be simultaneously opened is automatically computed by Arnold using a heuristic that tries to guess what the optimal number would be for each particular machine. We expect that the majority of users will be able to leave this at 0 and get the best performance.