



# JR Macros: Blender Filmic Transforms

## Introduction

This macro pack is designed to emulate Blender's *Filmic* view transform and looks, which provide an organic-looking tonal compression that makes renders instantly usable when saved out of Blender as non-linear gamma encoded images.

If you wish to work with OpenEXR or Radiance HDR exports, however, you may find it difficult to replicate these Filmic looks because Blender writes out purely linear unbounded float values to these formats.

Affinity Photo supports OpenColorIO, so you can of course load Blender's configuration into the app and use its view transforms. This does not write the transforms into the document colour values, however, so cannot be used to export to a non-linear format such as 8-bit JPEG, 16-bit TIFF etc.

There are workarounds to "bake" the Filmic Log transform and look (e.g. Very High Contrast) into the colour values, but it can be a complex procedure. It also requires having Blender's OpenColorIO configuration installed within Affinity Photo.

Instead, these macros offer a singleclick solution to get exactly the look you want, with *no dependency on OpenColorIO* at all. Even with a fresh install of Affinity Photo, just open your EXR/HDR render, find the macro library, then click to add whichever Filmic look you want.

😣 🗉 Library	≣.
▼ Default	Create New Category
Convert to sRGB (Steps: 2)	Import Macros
Strip metadata (Steps: 1)	
Black & White (Steps: 2)	
Flip Horizontal (Steps: 1)	
Flip Vertical (Steps: 1)	

#### Installation

- 1. Extract the *.afmacros* file to a directory of your choice.
- 2. In Affinity Photo, you will need to expose the **Library** panel. To do this, go to **View>Studio>Library**.
- 3. Click the small icon at the top right of the Library panel and choose Import Macros.
- 4. Navigate to the directory containing the *.afmacros* file and select it, then click **Open** (or double click the file).
- 5. The **Library** panel will then be populated with the macros from that category. If you are installing any other macro packs, repeat the process for those categories.

Tip: you can also drag-drop the afmacros file onto a blank area of the app and it will immediately import and be shown on the Library panel. You can bulk import multiple afmacros files this way.



### **Further Instructions**

*Please note*: when applying a Filmic transform macro for the first time after starting Affinity Photo, you may experience a delay of several seconds before the result appears. This is normal behaviour, as a complex 3D LUT is being loaded.

When applying these macros, the order of the layer stack is very important.

See the UI screenshot for guidance: you will want to add your Filmic transform *above* the render layers but *beneath* any additional adjustments, filters or other layer work you might be applying.

Also, you may need to adjust the overall exposure of your render if you have changed Blender's Exposure slider—some plugins, such as **Real Sky** and **Physical Starlight and Atmosphere** may do this automatically. Both of these plugins will reduce the Exposure to -6 stops, but this change is not reflected in the linear values of an EXR or HDR document when saved.

To emulate this, simply add an *Exposure* adjustment (Layer>New Adjustment Layer>Exposure) and place it *above* the render layers but *beneath* the Filmic transform group. On the Exposure adjustment dialog, change the slider to match the value used in Blender (e.g. -6).

If you prefer to do your own contrast grading, there is a Filmic Log (No Look) macro which only applies the log shaping and final sRGB device transform.



Finally, be aware that if you already have OpenColorIO configured within Affinity Photo, it will use the *OCIO Display Transform* method by default when you load an OpenEXR document. This must be changed back to ICC Display Transform to ensure consistency with the result you will get when exporting to 8/16-bit formats. You can access this option on the *32-bit Preview Panel* (View>Studio>32-bit Preview).





(Steps: 7) Low Contrast (Steps: 7) Medium Low Contrast

V JR - Blender Filmic Transforms

(Steps: 7) Medium Contrast

Very Low Contrast

Medium High Contrast (Steps: 6)

High Contrast (Steps: 7)

(Steps: 7)

Very High Contrast (Steps: 7)

Filmic Log (No Look) (Steps: 7)

**Display Transform** 

Unmanaged

ICC Display Transform

OCIO Display Transform

## Credits

All 3D renders and editing by James Ritson (http://www.jamesritson.co.uk)

Implementation for Affinity Photo derived from Blender's OpenColorIO configuration, supplied LUTs (for the "looks") and crosstalk 3D LUT. Blender's implementation is based on Troy Sobotka's Filmic Blender (<u>https://sobotka.github.io/filmic-blender/</u>).

Additional help from Andy Somerfield (Affinity Photo lead developer).

