

Supported Graphics Formats

Image Formats

This is the list of image file formats supported internally by Blender:

Format	Channel Depth	Alpha	Metadata	DPI	Extensions
BMP	8bit	✗	✗	✓	.bmp
Iris	8, 16bit	✓	✗	✗	.sgi .rgb .bw
PNG	8, 16bit	✓	✓	✓	.png
JPEG	8bit	✗	✓	✓	.jpg .jpeg
JPEG 2000	8, 12, 16bit	✓	✗	✗	.jp2 .j2k .j2c
Targa	8bit	✓	✗	✗	.tga
Cineon & DPX	8, 10, 12, 16bit	✓	✗	✗	.cin .dpx
OpenEXR	float 16, 32bit	✓	✓	✓	.exr
Radiance HDR	float	✓	✗	✗	.hdr
TIFF	8, 16bit	✓	✗	✓	.tif .tiff
WebP	8bit	✓	✓	✓	.webp

Hint

If you are not interested in technical details, a good rule of thumb for selecting output formats for your project is:

Use **OpenEXR**

if you intend to do compositing or color grading on these images.

Use **PNG**

if you intend on-screen output or encoding into multiple video formats.

Use **JPEG**

for on-screen output where file size is a concern and quality loss is acceptable.

All these formats support compression which can be important when rendering out animations.

Hint

Bit depths for image formats represent the following numbers of tonal levels per channel:

- 8: 256 levels
- 10: 1024 levels
- 12: 4096 levels
- 16: 65536 levels

Opening Images

Relative Path

Sets the file path to be relative to the currently opened blend-file.

See [Relative Paths](#).

Detect Sequences

Automatically looks for image sequences in the selected images (based on the file name). Disable this when you do want to get single images that are not part of a sequence. See [Opening an Image Sequence](#) for more information.

Detect UDIMs

Automatically looks for [UDIM](#) tiles in the directory of the selected image; if matches are found they are loaded into Blender as UDIMs. This works by detecting if the filename has a `.xxxx` (four digit number) before the file extension.

Opening an Image Sequence

To load image sequence in any of the supported image file formats, the filename of the images must contain a digit to indicate the frame order (e.g. `*-0001.jpg`, `*-0002.jpg`, `*-0003.jpg`, etc, of any image format), indicating the frame.

The sequence could be opened by the selection of the images with any of the following methods by the confirmation with the *Open Image* button or *Return*.

Range

Navigate into the directory and `LMB` click and drag over a range of names to highlight multiple files. You can page down and continue `Shift` - `LMB` click-dragging to add more to the selection.

Batch

`Shift` - `LMB` click selected non-related stills for batch processing; each image will be one frame, in sort order, and can be a mix of file types (`jpg`, `png`, `exr`, etc.).

All

Press `A` to select/deselect all files in the directory.

Saving Images

File Format

Choose what format to save the image as.

Color

The color format to save the image or video to. This setting is used by some formats to optimize how much data is written to the file. Note, *RGBA* not available for all image formats, check the list above for details.

BW:

Saves the image using grayscale colors.

RGB:

Saves red, green and blue channels

RGBA:

Saves red, green, blue and alpha channels.

Color Depth

The exponent value (with base two) for how many colors can be represented within a single color channel. A higher bit depth will allow more possible colors, reducing banding, and increasing precision. Yet a higher bit depth will increase memory usage exponentially.

Note, not all file formats support every color depth configuration. Below is a list of commonly used depths:

8-bit:

Most common for on-screen graphics and video.

10, 12, 16-bit:

Used for some formats focusing on photography and digital films (such as DPX and JPEG 2000).

16-bit Half Float:

Since full 32bit float is often more than enough precision, half float can save drive space while still providing a high dynamic range.

32-bit Float:

Highest quality color depth.

Note

Internally Blender's image system supports either:

- 8 bits per channel (4×8 bits).
- 32 bits float per channel (4×32 bits) – *using 4 times as much memory*.

Images higher than 8 bits per channel will be converted into a float on loading into Blender.

Compression

Used to reduce the size of the image file. How this is done may vary depending on the file format and settings used.

Quality

Controls the level of lossy compression applied to the image, expressed as a percentage. Lossy compression reduces file size by discarding some image data, which may result in a loss of detail.

- **0%**: Maximum compression, producing the smallest file size but the most noticeable quality loss.
- **100%**: No compression, preserving full image quality at the cost of a larger file size.

Save As Render

Save image with render [color management](#). For display image formats like PNG, apply view and display transform. For intermediate image format like OpenEXR, use the default render output color space.

Copy

Defines if the data-block will reference the newly created file or the reference will be unchanged, maintaining it with the original one.

Color Space

To specify the color space of the source file.

The list of color spaces depends on the active [OCIO config](#). The default supported color spaces are described in detail here: [Default OpenColorI Configuration](#)

Note

Note, *Cineon*, *DPX*, *OpenEXR*, and *Radiance HDR* image types default to being saved in a linear color space.

Format Details

Cineon & DPX

Cineon is Kodak's standard for film scanning, 10 bits per channel and logarithmic. DPX has been derived from Cineon as the ANSI/SMPTE industry standard. DPX supports 16-bit colors/channels, linear as well as logarithmic. DPX is currently a widely adopted standard used in the film hardware/software industry.

DPX as well as Cineon only stores and converts the "visible" color range of values between 0.0 and 1.0 (as a result of rendering or composite).

OpenEXR

[ILM's OpenEXR](#) has become a software industry standard for HDR image files, especially because of its flexible and expandable structure.

An OpenEXR file can store multiple layers and passes. This means OpenEXR images can be loaded into a Compositor keeping render layers and passes intact.

Output Options

Available options for OpenEXR render output are:

Color Depth

The exponent value (with base two) for how many colors can be represented within a single color channel. A higher bit depth will allow more possible colors, reducing banding, and increasing precision. Yet a higher bit depth will increase memory usage exponentially.

Float (Half):

Saves images in a custom 16 bits per channel in a floating-point format. This reduces the actual "bit depth" to 10-bit, with a 5-bit power value and 1-bit sign.

Float (Full):

Saves images using 32 bits per channel in a floating-point format.

Codec

The type of compression to encode the EXR-file with.

None:

Disables all compression for fastest encoding times but creates larger file sizes.

ZIP:

Lossless compression using Zlib on 16 row image blocks.

PIZ:

Lossless wavelet compression, effective for noisy/grainy images.

DWAA (lossy):

JPEG-like lossy compression on 32 row image blocks.

DWAB (lossy):

JPEG-like lossy compression on 256 row image blocks.

ZIPS:

Lossless compression using Zlib, each image row compressed separately

RLE:

Lossless run length encoding compression, works well when image rows have the same values.

Pxr24 (lossy):

Converts 32-bit floats to 24 bits then uses deflate compression. Pxr24 is lossless for half and 32-bit integer data and slightly lossy for 32-bit float data.

B44 (lossy):

Lossy compression for 16 bit float images, at fixed 2.3:1 ratio. B44 compresses uniformly regardless of image content.

B44A (lossy):

Lossy compression for 16 bit float images, at fixed 2.3:1 ratio with further compression on areas of flat color are further compressed, such as alpha channels.

Quality DWAA (lossy) DWAB (lossy)

Controls the level of lossy compression applied to the image, expressed as a percentage. Lossy compression reduces file size by discarding some image data, which may result in a loss of detail.

- **0%:** Maximum compression, producing the smallest file size but the most noticeable quality loss.
- **100%:** No compression, preserving full image quality at the cost of a larger file size.

Preview

When rendering animations (or single frames via command line), save a JPEG copy of the image, for a quick preview.

Radiance HDR

Radiance is a suite of tools for lighting simulation. Since Radiance had the first (and for a long time the only) HDR image format, this format is supported by many other software packages.

Radiance `.hdr` files store colors still in 8 bits per component, but with an additional (shared) 8-bit exponent value, making it 32 bits per pixel.

[Previous Media Formats](#)

Copyright © : This page is licensed under a CC-BY-SA 4.0 Int. License

Made with [Furo](#)

Last updated on 2025-05-10

[View Source](#)
[View Translation](#)
[Report issue on this page](#)

[Not Supported Video & Audio Format](#)