Contents

SurfaceLevel 2.0

Overview

SurfaceLevel 2.0 is a texture converter and basic image editor. It loads many image file formats, converts to and from almost every video-game API texture (OpenGL, Vulkan, Metal, and Direct3D 12), offers a fast and high-quality resampler, and saves to many file formats. It is useful for performing nearly all video-game–related texture operations, such as compression, pre-multiplying alpha, generating normal maps, resampling/generating crisp mipmaps, etc., and for general-purpose image resampling, colorspace conversion, format conversion, etc. SurfaceLevel 2.0 is meant to gather a large quantity of features together and to provide more options for those features than you will find in any other tool. For example, when resampling, it is possible to specify different samplers for your image width, height, and depth, and different samplers for the RGB colors and the alpha channel—the sharpest resamplers often create ringing, which is particularly bad for the alpha channel; here you can using a ringing filter for the colors but a non-ringing filter for alpha (while specifying the same or different filters for both the color and alpha widths, heights, and depths).

SurfaceLevel 2.0 seeks to be useful in the level of detail for each feature, the number of features, and the performance of each feature. Supports volume (3-D) textures, texture arrays, cube maps, and mipmaps.

Control Flow

SurfaceLevel 2.0 doesn’t try to be overly smart, so as long as you understand the basic internal workflow it should be easy to predict what it will do in any edge cases. Internally it performs the following operations in order:

1. Loads the image file.
2. Performs a conversion to the desired target format.
    a. Converts to RGBA64F.
    b. Crops.
    c. Converts to linear, applying any applicable colorspace conversions and gamma corrections necessary.
    d. Applies requested transforms, such as swapping, swizzling, flipping, etc.
    e. Performs resampling and generates mipmaps.
    f. Applies pre-multiplied alpha.
    g. Converts to the requested texture format, applying gamma-correction as necessary, etc.
3. Saves to the desired file.
    a. If the current format is not directly supported by the file, it is converted to the nearest format that is supported by the file.

The ways it tries to be smart:

1. By default, it will try to ensure only 1 gamma transform occurs from source to linear and from linear to destination. If images do not contain gamma data, they are assumed to be sRGB (`-g`, `-gamma`, `-srgb`, `-rgbe`, and `-linear` to change this assumption). If they contain gamma data from an embedded ICC profile, that is used instead. However, if you manually supply a source gamma value with `-g`, `-gamma`, `-srgb`, `-rgbe`, or `-linear`, your supplied gamma will be stacked with any contained/embedded gamma data. This can allow you to correct images that may have been saved with incorrect gamma. To specify your own gamma curve to be used *in-place* of any embedded or selected colorspace profiles, use `-g`, `-gamma`, `-srgb`, `-rgbe`, or `-linear` to define your own gamma curve and `-ignore_input_colorspace_gamma` to ignore the gamma curve in any colorspace profiles being used.
2. To simplify the process, it is not necessary to specify the export format. If your conversion format is not supported, it will find the closest match that is supported. It will try to ensure a lossless conversion, but the option to specify the export format is always available if needed. This only applies to general image formats. For specialized GPU-leaning formats, such as DDS, KTX, PVR, etc., export will fail if the format specified in `-format` is not supported by the file.
   For example, `-png_format` is never strictly necessary because any format supplied by `-format` can be automatically converted to a format supported by the PNG file specification; `-png_format` is entirely optional. However, no automatic

conversion is made when saving to DDS, for example, so the format supplied by –`format` must be supported by the DDS file specification.

Commands

File

Command	Parameter	Description
-file	<file path>	A path to an image file to load and convert. The **-file**, **-yuv_file**, and **-outfile** commands can be used multiple times to load and save multiple files.
-yuv_file	<file path> <width> <height>	Path to a YUV file to load. If the extension does not indicate the YUV encoding, then **-yuv_input_format** must be called to specify the YUV encoding. Recognized file extensions: yuv444p16 yuv444p12le yuv444p10le yuv444p yuv444y16 yuv444y12le yuv444y10le yuv444y yuv422p16 yuv422p12le yuv422p10le yuv422p yuv422y16 yuv422y12le yuv422y10le yuv422y yuv420p16 yuv420p12le yuv420p10le yuv420p yuv420y16 yuv420y12le yuv420y10le yuv420y yuva12le yuva10le

		uyvy16 uyvy12le uyvy10le uyvy y210 yuy2 yv12 yv12 p016 p010 y012 y010 nv12 nv21 y416 y410 ayuv Extensions other than these will require the format to be explicitly set.
-outfile **-out_file**	<file path>	The path to which to save the file supplied with the last **-file** command. The destination file format is determined by the file extension. Currently supported formats: PNG BMP TGA JPG J2K JP2 EXR DDS KTX PVR PBM PGM ICO

Gamma/Colorspaces

Command	Parameter	Description
-gamma **-g**	<gamma>	Sets the input gamma power. Defaults to **-2.2**

		(precise sRGB). See Notes.
-targetgamma	<gamma>	Sets the output gamma power. Defaults to **-2.2** (precise sRGB).
-rgbe **-linear**		Sets the source and output gamma to 0.0.
-srgb		Sets the source gamma to -2.2 (precise sRGB).
-target_srgb		Sets the output gamma to -2.2 (precise sRGB).
-input_colorspace	sRGB sRGB_precise	Sets the source colorspace profile to an accurate no-gap sRGB.
	sRGB_std sRGB_standard	Sets the source colorspace profile to the standard sRGB.
	smpte_170 smpte_170m 170m	Sets the source colorspace profile to an accurate no-gap SMPTE 170M-1999.
	smpte_170_std smpte_170m_std 170m_std smpte_170_standard smpte_170m_standard 170m_standard	Sets the source colorspace profile to the standard SMPTE 170M-1999.
	rec709 rec.709 bt709 bt.709 itu_bt709 itu_bt.709	Sets the source colorspace profile to an accurate no-gap ITU-R Recommendation BT.709-5.
	rec709_std rec.709_std bt709_std bt.709_std itu_bt709_std itu_bt.709_std rec709_standard rec.709_standard bt709_standard	Sets the source colorspace profile to the standard ITU-R Recommendation BT.709-5.

	bt.709_standard itu_bt709_standard itu_bt.709_standard	
	adobe adobergb adobe_rgb	Sets the source colorspace profile to Adobe RGB (1998) Color Image Encoding Version 2005-05.
	bt2020 bt.2020 itu_bt2020 itu_bt.2020	Sets the source colorspace profile to an accurate no-gap ITU-R Recommendation BT.2020.
	bt2020_std bt.2020_std itu_bt2020_std itu_bt.2020_std bt2020_standard bt.2020_standard itu_bt2020_standard itu_bt.2020_standard	Sets the source colorspace profile to the standard ITU-R Recommendation BT.2020.
	dcip3 dci-p3 dci_p3	Sets the source colorspace profile to SMPTE RP 431-2:2011.
	smpte_240 smpte_240m 240m	Sets the source colorspace profile to an accurate no-gap SMPTE 240M-1999.
	smpte_240_std smpte_240m_std 240m_std smpte_240_standard smpte_240m_standard 240m_standard	Sets the source colorspace profile to the standard SMPTE 240M-1999.
	ntsc_1953 ntsc1953	Sets the source colorspace profile to an accurate no-gap NTSC 1953.
	ntsc_1953_std ntsc1953_std ntsc_1953_standard ntsc1953_standard	Sets the source colorspace profile to the standard NTSC 1953.

	tech_3213 tech3213	Sets the source colorspace profile to an accurate no-gap EBU Tech. 3213.
	tech_3213_std tech3213_std tech_3213_standard tech3213_standard	Sets the source colorspace profile to the standard EBU Tech. 3213.
	displayp3 display-p3 display_p3	Sets the source colorspace profile to an accurate no-gap Display P3 Color Encoding (v 1.0).
	displayp3_std display-p3_std display_p3_std displayp3_standard display-p3_standard display_p3_standard	Sets the source colorspace profile to the standard Display P3 Color Encoding (v 1.0).
	rec601 rec.601 bt601 bt.601 itu_bt601 itu_bt.601	Sets the source colorspace profile to an accurate no-gap ITU-R Recommendation BT.601 (525).
	rec601_std rec.601_std bt601_std bt.601_std itu_bt601_std itu_bt.601_std rec601_standard rec.601_standard bt601_standard bt.601_standard itu_bt601_standard itu_bt.601_standard	Sets the source colorspace profile to the standard ITU-R Recommendation BT.601 (525).
	rec601_pal rec.601_pal bt601_pal bt.601_pal itu_bt601_pal itu_bt.601_pal	Sets the source colorspace profile to an accurate no-gap ITU-R Recommendation BT.601 (625).

	rec601_pal_std rec.601_pal_std bt601_pal_std bt.601_pal_std itu_bt601_pal_std itu_bt.601_pal_std rec601_pal_standard rec.601_pal_standard bt601_pal_standard bt.601_pal_standard itu_bt601_pal_standard itu_bt.601_pal_standard	Sets the source colorspace profile to the standard ITU-R Recommendation BT.601 (625).
	generic_film film	Sets the source colorspace profile to generic film.
	bt470_ntsc bt470_m_ntsc	Sets the source colorspace profile to an accurate no-gap Rec. ITU-R BT.470-6 (M/NTSC).
	bt470_pal bt470_m_pal	Sets the source colorspace profile to an accurate no-gap Rec. ITU-R BT.470-6 (M/PAL).
	bt470_b bt470_b1 bt470_d bt470_d1 bt470_g bt470_h bt470_k bt470_k1 bt470_l bt470_n_pal bt470_secam bt470_l_secam	Sets the source colorspace profile to an accurate no-gap Rec. ITU-R BT.470-6 (B, B1, D, D1, G, H, K, N/PAL, K1, L/SECAM).
	ntsc_1987 smpte_c	Sets the source colorspace profile to SMPTE C with a pow(2.2) curve.
	ntsc_1987_std smpte_c_std	Sets the source colorspace profile to the standard SMPTE C.

	romm_rgb rommrgb	Sets the source colorspace profile to Reference Output Medium Metric RGB (ROMM RGB).
	rimm_rgb rimmrgb	Sets the source colorspace profile to Reference Input Medium Metric RGB (RIMM RGB).
	erimm_rgb erimmrgb	Sets the source colorspace profile to Extended Reference Input Medium Metric RGB (ERIMM RGB).
	plasa plasa_ansi	Sets the source colorspace profile to PLASA ANSI E1.54.
	protune gopro	Sets the source colorspace profile to Protune Native (GoPro).
	s-gamut sgamut s_gamut	Sets the source colorspace profile to S-Gamut.
	s-gamut3 sgamut3 s_gamut3	Sets the source colorspace profile to S-Gamut3.
	s-gamut3cine sgamut3cine s_gamut3cine s-gamut3_cine sgamut3_cine s_gamut3_cine	Sets the source colorspace profile to S-Gamut3.Cine.
-target_colorspace	Same as for **-input_colorspace**.	Sets the output colorspace profile.
-input_colorspace_file	<file path>	Sets the input colorspace profile. Loads .ICC and .ICM files.
-target_colorspace_file	<file path>	Sets the output colorspace profile. Loads .ICC and .ICM files.

-dont_embed_icc		No colorspace profile will be embedded into files with colorspace-profile support.
-embed_icc		Any specified output colorspace profiles will be embedded into files with colorspace-profile support. This is the default.
-ignore_input_colorspace_gamma		The gamma in any supplied or embedded input colorspace profile will be ignored.
-rendering_intent **-render_intent**	perceptual	All colors are scaled to fit into the target colorspace. Useful for converting from wide colorspaces to more narrow ones.
	relative_colorimetric	Colors in gamut are unchanged, but colors outside of gamut are clipped to the nearest in-gamut color. This is the default.
	saturation	Like **perceptual**, but tends to make colors more saturated.
	absolute_colorimetric	Not intended for color conversion, but rather typically for digital inkjet proofing.

Resampling

Command	Parameter	Description
-filter	box point	Applies the selected filter to all non-mipmap filters.
	tent linear	The default mipmap alpha filter.
	quadraticsharp quadratic_sharp	The default non-mipmap filter.
	quadratic	

	quadraticapprox quadraticapproximate quadratic_approximate	
	quadraticmix quadratic_mix	
	kaiser	
	lanczos2	
	lanczos3	
	lanczos4	
	lanczos6	
	lanczos8	
	lanczos12	
	lanczos64	
	mitchell	One of the best choices for upscaling.
	catmul catmulrom catmul_rom catmul-rom	
	bspline b-spline b_spline	
	cardinal card cardinaluniform cardinal_uniform	The default mipmap color filter.
	hermite	
	hamming	
	hanning	
	blackman	
	gaussiansharp gaussian_sharp	
	gaussian	
	bell	
-filtera **-filter_alpha**	Same as for **-filter**.	Sets the non-mipmap alpha *width*, *height*, and *depth* filter.
-filterw	Same as for **-filter**.	Sets the non-mipmap color and alpha *width* filter.
-filterh	Same as for **-filter**.	Sets the non-mipmap color and alpha *height* filter.
- filterd	Same as for **-filter**.	Sets the non-mipmap color and alpha *depth* filter.

-filterw_color	Same as for **-filter**.	Sets the non-mipmap color *width* filter.
-filterh_color	Same as for **-filter**.	Sets the non-mipmap color *height* filter.
-filterd_color	Same as for **-filter**.	Sets the non-mipmap color *depth* filter.
-filterw_alpha	Same as for **-filter**.	Sets the non-mipmap alpha *width* filter.
-filterh_alpha	Same as for **-filter**.	Sets the non-mipmap alpha *height* filter.
-filterd_alpha	Same as for **-filter**.	Sets the non-mipmap alpha *depth* filter.
-prescale	<new width> <new height>	Resamples the image to the given width/height using the selected non-mipmap filters.
-prescale3 **-resample_size**	<new width> <new height> <new depth>	Resamples the image to the given width/height/depth using the selected non-mipmap filters.
-resample_to	nearest	Resamples to the nearest power of 2 in each dimension.
	lo	Resamples to the next power-of-2 down.
	hi	Resamples to the next power-of-2 up.
-rel_scale	<width multiplier> <height multiplier>	Resamples by the given width and height multipliers.
-rel_scale3	<width multiplier> <height multiplier> <depth multiplier>	Resamples by the given width, height, and depth multipliers.
-clamp2 **-clamp**	<width> <height>	Clamps the image to the given width and height.
-clamp3	<width> <height> <depth>	Clamps the image to the given width, height, and depth.

Texture Addressing

Command	**Parameter**	**Description**
-textureaddressing **-ta**	clamp	U, V, and W coordinates are clamped to the edge of the texture.

		Equal to `D3D12_TEXTURE_ADDRESS_MODE_CLAMP`.
	repeat wrap	U, V, and W coordinates repeat beyond the 0..1 range. Equal to `D3D12_TEXTURE_ADDRESS_MODE_WRAP`.
	mirror reflect	U, V, and W are mirrored beyond the 0..1 range. Equal to `D3D12_TEXTURE_ADDRESS_MODE_MIRROR`.
	mirroronce mirror_once	U, V, and W are mirrored 1 time beyond the 0..1 range, after which clamping is used. Equal to `D3D12_TEXTURE_ADDRESS_MODE_MIRROR_ONCE`.
	border bordercolor border_color	The border color is used when U, V, and W go outside of 0..1. Equal to `D3D12_TEXTURE_ADDRESS_MODE_BORDER`.
	no_border nul_border	Nothing is considered to exist beyond the U, V, and W texture edges. This is the default.
-textureaddressingw **-taw**	Same as **-ta**.	Applies only to the U coordinate.
-textureaddressingh **-tah**	Same as **-ta**.	Applies only to the V coordinate.
-textureaddressingd **-tad**	Same as **-ta**.	Applies only to the W coordinate.
-textureaddressingw_opaque **-taw_color**	Same as **-ta**.	Applies only to the U coordinate and to color channels.
-textureaddressingh_opaque **-tah_color**	Same as **-ta**.	Applies only to the V coordinate and to color channels.
-textureaddressingd_opaque **-tad_color**	Same as **-ta**.	Applies only to the W coordinate and to color channels.
-textureaddressingw_alpha	Same as **-ta**.	Applies only to the U coordinate and to the alpha channel.

-taw_alpha		
-textureaddressingh_alpha **-tah_alpha**	Same as **-ta**.	Applies only to the V coordinate and to the alpha channel.
-textureaddressingd_alpha **-tad_alpha**	Same as **-ta**.	Applies only to the W coordinate and to the alpha channel.
-border_color	<r> <g> <b> <a>	Sets the border color for the U, V, and W coordinates. Defaults to **0.0 0.0 0.0 1.0**.

Cropping

Command	Parameter	Description
-crop	<x> <y> <width> <height>	Crops the input image to the 2-D area specified. Depth/volume images will retain their depths. Cropping outside of the image area is allowed. How areas outside the image area are handled depends on the color texture addressing modes, set via the **-textureaddressing** commands.
-crop3	<x> <y> <z> <width> <height> <depth>	Crops a 3-D volume/depth image.
-bake_tex_mapping_u	<address mode> <repeats>	Bakes a texture-addressing mode into a texture’s U texture coordinates. The addressing mode is one of the **-textureaddressing** values, and *<repeats>* indicates how many copies to the left and right to make of the original image. Each copy will be repeated, mirrored, clamped, or border-color’d, which allows baking the texture-addressing into the texture for systems that don’t support a given addressing mode or combination of different UVW addressing modes.
-bake_tex_mapping_v	<address mode> <repeats>	Bakes a texture-addressing mode into a texture’s V texture

		coordinates. The addressing mode is one of the **-textureaddressing** values, and <*repeats*> indicates how many copies to the top and bottom to make of the original image.
-bake_tex_mapping_w	<address mode> <repeats>	Bakes a texture-addressing mode into a texture's W texture coordinates. The addressing mode is one of the **-textureaddressing** values, and <*repeats*> indicates how many copies to the front and back to make of the original image.

YUV Options

Command	Parameter	Description
-yuv_input_format	<any Vulkan, DXGI, or Metal YUV format>	Sets the format (encoding) of the YUV file being loaded.
	nv12	DXGI_FORMAT_NV12/ VK_FORMAT_G8_B8R8_2PLANE_420_UNORM
	nv21	DXGI_FORMAT_NV21
	yv12	DXGI_FORMAT_YV12
	yuy2	DXGI_FORMAT_YUY2/VK_FORMAT_G8B8G8R8_422_UNORM/DXGI_FORMAT_G8R8_G8B8_UNORM/MTLPixelFormatGBGR422
	uyvy	DXGI_FORMAT_R8G8_B8G8_UNORM/ VK_FORMAT_B8G8R8G8_422_UNORM/ MTLPixelFormatBGRG422
	p010	DXGI_FORMAT_P010/ VK_FORMAT_G10X6_B10X6R10X6_2PLANE_420_UNORM_3PACK16
	p016	DXGI_FORMAT_P016/ VK_FORMAT_G16_B16R16_2PLANE_420_UNORM
	p210	DXGI_FORMAT_P210/ VK_FORMAT_G10X6_B10X6R10X6_2PLANE_422_UNORM_3PACK16
	p216	DXGI_FORMAT_P216/ VK_FORMAT_G16_B16R16_2PLANE_422_UNORM

	y210	DXGI_FORMAT_Y210/VK_FORMAT_G10X6B10X6G10X6R10X6_422_UNORM_4PACK16
	y216	DXGI_FORMAT_Y216/VK_FORMAT_G16B16G16R16_422_UNORM
	y410	DXGI_FORMAT_Y410
	y416	DXGI_FORMAT_Y416
	ayuv	DXGI_FORMAT_AYUV
-yuv_format	Same as **-yuv_input_format**.	Sets the encoding when saving to a YUV format.
-yuv_input_use_approx		Uses a common approximate YUV -> RGB conversion when loading a YUV file. By default the full YUV -> RGB algorithm is used.
-yuv_use_approx		Uses a common approximate RGB -> YUV conversion when writing to a YUV file. By default the full RGB -> YUV algorithm is used.
-yuv_input_kr_kb	REC_709 REC709	Sets the Kr and Kb factors according to the ITU-R Recommendation BT.709-5 standard. 0.212639005871510 and 0.072192315360734. This is the default.
	REC_2020 REC2020	Sets the Kr and Kb factors according to the ITU-R Recommendation BT.2020 standard. 0.2627 and 0.0593.
	SMPTC	Sets the Kr and Kb factors according to the SMPTE C standard. 0.2124 and 0.0866.
	REC_601 REC601	Sets the Kr and Kb factors according to the ITU-R Recommendation BT.601 standard. 0.2988390 and 0.1143500.
	CIE_1931 CIE1931	Sets the Kr and Kb factors according to the CIE 1931 standard. 0.3086 and 0.0820.
	NTSC_1953 NTSC1953	Sets the Kr and Kb factors according to the NTSC 1953 standard. 0.3 and 0.11.
	EBU_TECH_3213 EBUTECH3213	Sets the Kr and Kb factors according to the EBU Tech. 3213 standard. 0.2988390 and 0.1143500.

-yuv_kr_kb	Same as **-yuv_input_kr_kb**.	Sets the output Kr and Kb factors when saving to a YUV file.
-yuv_input_set_kr_kb	<kr> <kb>	Manually specifies the Kr and Kb factors for loading a YUV file.
-yuv_set_kr_kb	<kr> <kb>	Manually specifies the Kr and Kb factors for saving to a YUV file.
-yuv_input_set_z -yuv_input_set_black	<black level>	Sets the black level (0..1) for loading a YUV file. Defaults to **0.0**.
-yuv_set_z -yuv_set_black	<black level>	Sets the black level (0..1) for saving to a YUV file.
-yuv_input_set_s -yuv_input_set_scale	<scale>	Sets the scaler (0..1) for loading a YUV file. Defaults to **1.0**.
-yuv_set_s -yuv_set_scale	<scale>	Sets the scaler (0..1) for saving to a YUV file. Defaults to **1.0**.
-yuv_input_pc		Sets the black level to 0.0, scale to 1.0, Kr and Kb to the ITU-R Recommendation BT.709-5 standard, and enables the full non-approximate conversion routine for loading YUV files.
-yuv_pc		Sets the black level to 0.0, scale to 1.0, Kr and Kb to the ITU-R Recommendation BT.709-5 standard, and enables the full non-approximate conversion routine for saving to a YUV file.
-yuv_input_studio		Sets the black level to (16.0 / 255.0), scale to (219.0 / 255.0), Kr and Kb to the ITU-R Recommendation BT.709-5 standard, and enables the full non-approximate conversion routine for loading YUV files.
-yuv_studio		Sets the black level to (16.0 / 255.0), scale to (219.0 / 255.0), Kr and Kb to the ITU-R Recommendation BT.709-5 standard, and

		enables the full non-approximate conversion routine for saving to a YUV file.

Indices & Palettes

Command	Parameter	Description
-gen_pal **-gen_palette**		Generates a new palette for indexed images.
-gen_pal_iterations	<iterations>	Sets the maximum number of iterations when generating a palette. The higher the better. Defaults to the number of colors in the palette. In practice, the number of colors in a palette serves as a hard maximum, but it will exit early if iterations stop causing refinements.
-pal_dither	floyd floyd-steinburg	Selects the type of dithering to use for palettes. Implements Floyd-Steinburg Dithering. This is the default.
	jjn	Implements Jarvis, Judice, and Ninke Dithering.
	stucki	Implements Stucki Dithering, presented 5 years after JJN, offering a similar dither with a slight performance improvement.
	burkes	Implements Burkes Dithering, presented 7years after Stucki, offering yet-another minor hit to quality in exchange for performance.
	sierra	Implements Sierra Dithering.
	sierra2row	Implements Two-Row Sierra Dithering.
	sierralite sierra_lite	Implements Sierra Lite Dithering.
	atkinson atk	Implements Atkinson Dithering, used by the original Macintosh computer.
	bayer4 bayer4x4	Implements a 4×4 Bayer Dither.

	bayer8 bayer8x8	Implements an 8×8 Bayer Dither.
-dither_error_weight **-dither_error_weights**	<r> <g> <b> <a>	Sets the per-channel weights for dithering. Defaults to **0.925 0.925 0.925 1.0**.
-dither_error_weight_full **-dither_error_weight_100**		Sets the dithering weights to 1.0 1.0 1.0 1.0.
-dither_error_weight_75		Sets the dithering weights to 0.75 0.75 0.75 1.0.
-dither_error_weight_half **-dither_error_weight_50**		Sets the dithering weights to 0.5 0.5 0.5 1.0.
-dither_error_weight_25		Sets the dithering weights to 0.25 0.25 0.25 1.0.
-dither_error_weight_perceptual **-dither_error_weight_perc**	REC_709 REC709	Sets the dithering weights to 0.212639005871510 0.715168678767756 0.072192315360734.
	REC_2020 REC2020	Sets the dithering weights to 0.2627 0.678 0.0593.
	SMPTC	Sets the dithering weights to 0.2124 0.7011 0.0866.
	REC_601 REC601 EBU_TECH_3213 EBUTECH3213	Sets the dithering weights to 0.2988390 0.5868110 0.1143500.
	CIE_1931 CIE1931	Sets the dithering weights to 0.3086 0.6094 0.0820.
	NTSC_1953 NTSC1953	Sets the dithering weights to 0.3 0.59 0.11.
-dither_error_weight_scale	<scale>	Scales the dithering weights by the given amount. Can be used to strengthen or soften the effects of perceptual weights. Weights are multiplied each time this command is encountered.

Mipmaps

Command	Parameter	Description
-nomips **-nomipmaps** **-no_mips** **-no_mipmaps**		No mipmaps are generated and existing mipmaps are discarded.

-nmips	<total mipmaps>	Sets the total number of mipmaps desired. Set to 0 to generate (or keep) a full mipmap chain. Defaults to **0**.
-keepmips **-keepmipmaps** **-keep_mips** **-keep_mipmaps**		By default, new mipmaps will be generated as specified by **-nmips** (which defaults to **0**, so a full chain is generated). This command allows existing mipmaps to be retained instead of overwritten. Existing mipmaps beyond the **-nmips** specification will be truncated, and if **-nmips** extends beyond the number of existing mipmaps then new mipmaps will be generated to fill the gap.
-mip_filter	Same as for **-filter**.	Applies the selected filter to all mipmap filters.
-mip_filtera **-mip_filter_alpha**	Same as for **-filter**.	Sets the mipmap alpha *width*, *height*, and *depth* filter.
-mip_filterw	Same as for **-filter**.	Sets the mipmap color and alpha *width* filter.
-mip_filterh	Same as for **-filter**.	Sets the mipmap color and alpha *height* filter.
-mip_filterd	Same as for **-filter**.	Sets the mipmap color and alpha *depth* filter.
-mip_filterw_color	Same as for **-filter**.	Sets the mipmap color *width* filter.
-mip_filterh_color	Same as for **-filter**.	Sets the mipmap color *height* filter.
-mip_filterd_color	Same as for **-filter**.	Sets the mipmap color *depth* filter.
-mip_filterw_alpha	Same as for **-filter**.	Sets the mipmap alpha *width* filter.
-mip_filterh_alpha	Same as for **-filter**.	Sets the mipmap alpha *height* filter.
-mip_filterd_alpha	Same as for **-filter**.	Sets the mipmap alpha *depth* filter.

Normal Maps

Command	Parameter	Description
-nm_channel	r red	The normal map will be generated using the R channel.
	g green	The normal map will be generated using the G channel.
	b blue	The normal map will be generated using the B channel.
	a alpha	The normal map will be generated using the A channel.

	max	The normal map will be generated using the max value between the RGBA channels. This is the default.
	rgb	The normal map will be generated using the average value between the RGB channels.
	colorspace	The normal map will be generated using the weighted average value between the RGB channels. Use **-luma** to select from predefined weights or **–weight** to manually specify weights.
-norm **-normalize**		Indicates that the normal maps should be normalized.
-opengl **-unity** **-blender** **-maya**		Specifies that the normal map should be compatible with OpenGL.
-directx **-ue4** **-unreal** **-unrealengine** **-ue** **-dsmax**		Specifies that the normal map should be compatible with DirectX.
-n3x3		Normal maps will be generated with a 3×3 kernel.
-n5x5		Normal maps will be generated with a 5×5 kernel.
-n7x7		Normal maps will be generated with a 7×7 kernel.
-n9x9		Normal maps will be generated with a 9×9 kernel.
-scale **-nm_z**		Specifies the normal map's Z influence.

Transforms

Command	Parameter	Description
-format	<any Vulkan, DXGI, or Metal format>	Converts the loaded image to the given texture format.
-ogl_format	<internal format> <type> <base internal format>	Converts the loaded image to the given OpenGL texture format.
-ignore_alpha		Any alpha channel is set to all 1's.

-alpha_threshold	<cutoff>	Sets the alpha cutoff ([0..255]) for conversions to formats with binary alpha. Defaults to **128**.
-premultiply_alpha **-premult_alpha**		Specifies that alpha should be pre-multiplied. If an image is already pre-multiplied it is not pre-multiplied again.
-swizzle	<swizzle>	Specifies a swizzle to apply. Valid swizzle characters: rgbaxyzw01. Must be 4 characters long and is not case-sensitive.
-swap		Swaps the R and B channels.

Quality Settings

Command	Parameter	Description
-quality_highest **-very_slow**		The highest quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.
-quality_production **-slow**		The 2nd-highest quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.
-quality_normal **-basic**		The normal quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.
-fast		A fast but somewhat low-quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.
-quick **-veryfast**		A faster but lower-quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.
-ultrafast		The fastest but lowest-quality setting for compressing textures in BC*, EAC, ETC*, PVR, and ASTC formats.

Misc.

Command	Parameter	Description
-weight **-weights**	<red weight> <green weight> <blue weight>	Sets the luminance weight factors manually.
-luma	REC_709 REC709	Sets the luminance weight factors according to the ITU-R Recommendation BT.709-5 standard. 0.212639005871510,

		0.715168678767756, and 0.072192315360734. This is the default.
	REC_2020 REC2020	Sets the luminance weight factors according to the ITU-R Recommendation BT.2020 standard. 0.2627, 0.678, and 0.0593.
	SMPTC	Sets the luminance weight factors according to the SMPTE C standard. 0.2124, 0.7011, and 0.0866.
	REC_601 REC601	Sets the luminance weight factors according to the ITU-R Recommendation BT.601 standard. 0.2988390, 0.5868110, and 0.1143500.
	CIE_1931 CIE1931	Sets the luminance weight factors according to the CIE 1931 standard. 0.3086, 0.6094, and 0.0820.
	NTSC_1953 NTSC1953	Sets the luminance weight factors according to the NTSC 1953 standard. 0.3, 0.59, and 0.11.
	EBU_TECH_3213 EBUTECH3213	Sets the luminance weight factors according to the EBU Tech. 3213 standard. 0.2988390, 0.5868110, and 0.1143500.
-printformats **-print_formats**		Prints all supported formats that can be supplied to **-format**.

PNG Options

Command	Parameter	Description
-png_default		Default PNG compression (6) will be used.
-png_bestspeed		Fast PNG compression (1) will be used.
-png_bestcompression		Best PNG compression (9) will be used.
-png_level		Specifies the PNG compression level. [0..9].
-png_nocompression		No PNG compression will be used.
-png_interlaced		Interlacing will be used. The default is no interlacing.

-png_format	R8G8B8 RGB24 RGB	Specifies the PNG format to which to save. If the format is not specified, the closest format to what was specified by **-format** (or the original file’s format if **-format** is not specified) will be used.
	R8G8B8_SRGB RGB24_SRGB RGB_SRGB	
	R8G8B8A8 RGBA32 RGBA	
	R8G8B8A8_SRGB RGBA32_SRGB RGBA_SRGB	
	R16G16B16 RGB16	
	R16G16B16A16 RGBA16	
	L8 LUMINANCE8 GREYSCALE8 GRAYSCALE8	
	L16 LUMINANCE16 GREYSCALE16 GRAYSCALE16	
	I1 INDEXED1	
	I2 INDEXED2	
	I4 INDEXED4	
	I8 INDEXED8	

BMP Options

Command	Parameter	Description

-bmp_rle		RLE encoding will be used to save the BMP file. This is not set by default.
-bmp_noalpha		Alpha will not be saved or will be set to 1 in the BMP file. The default is to store alpha when available.
-bmp_nobitmask **-bmp_nomask**		By default, the BMP file will contain masks for the R, G, B, and A channels. This setting causes the BMP file to be saved without the masks. See Notes.
-bmp_format	R8G8B8 RGB24 RGB	Specifies the BMP format to which to save. See Notes. If the format is not specified, the closest format to what was specified by **-format** (or the original file's format if **-format** is not specified) will be used.
	R8G8B8_SRGB RGB24_SRGB RGB_SRGB	
	R8G8B8A8 RGBA32 RGBA	
	R8G8B8A8_SRGB RGBA32_SRGB RGBA_SRGB	
	B8G8R8A8	
	B8G8R8A8_SRGB	
	A8B8G8R8	
	A8B8G8R8_SRGB	
	R4G4B4A4	
	B4G4R4A4	
	A4R4G4B4	
	A4B4G4R4	
	R5G6B5	

	B5G6R5	
	R5G5B5A1	
	A1B5G5R5	
	A1R5G5B5	
	A4B4G4R4	

TGA Options

Command	Parameter	Description
-tga_rle		The TGA file will be saved with RLE encoding.
-tga_format	R8G8B8 RGB24 RGB	Specifies the TGA format to which to save. If the format is not specified, the closest format to what was specified by **-format** (or the original file's format if **-format** is not specified) will be used.
	R8G8B8_SRGB RGB24_SRGB RGB_SRGB	
	R8G8B8A8 RGBA32 RGBA	
	R8G8B8A8_SRGB RGBA32_SRGB RGBA_SRGB	
	A1R5G5B5 A1RGB5	
	L8 LUMINANCE8 GREYSCALE8 GRAYSCALE8	

EXR Options

Command	Parameter	Description
-exr_float		Saves data as float instead of as half (not recommended).
-exr_zip		Saves with zlib compression in blocks of 16 scan lines.
-exr_piz		Saves with piz-based wavelet compression.

-exr_pxr24		Saves with lossy 24-bit float compression.
-exr_b44		Saves with lossy 44% float compression—goes to 22% when combined with **-exr_lc**.
-exr_lc		Saves images with one luminance and two chroma channels, rather than as RGB (lossy compression).

J2K Options

Command	Parameter	Description
-j2k_comp **-j2k_compression**	<X>	Quality level (1..512). Save rate is X:1. Default is 16:1.
-j2k_format	R8G8B8 RGB24 RGB	Specifies the J2K format to which to save. If the format is not specified, the closest format to what was specified by **-format** (or the original file's format if **-format** is not specified) will be used.
	R8G8B8_SRGB RGB24_SRGB RGB_SRGB	
	R8G8B8A8 RGBA32 RGBA	
	R8G8B8A8_SRGB RGBA32_SRGB RGBA_SRGB	
	L16 LUMINANCE16 GREYSCALE16 GRAYSCALE16	
	R16G16B16 RGB16	
	R16G16B16A16 RGBA16	

JP2 Options

Command	Parameter	Description
-jp2_comp **-jp2_compression**	<X>	Quality level (1..512). Save rate is X:1. Default is **16**:1.
-jp2_format	R8G8B8 RGB24 RGB	Specifies the JP2 format to which to save. If the format is not specified, the closest format to what was specified by **-format** (or

		the original file’s format if **-format** is not specified) will be used.
	R8G8B8_SRGB RGB24_SRGB RGB_SRGB	
	R8G8B8A8 RGBA32 RGBA	
	R8G8B8A8_SRGB RGBA32_SRGB RGBA_SRGB	
	L16 LUMINANCE16 GREYSCALE16 GRAYSCALE16	
	R16G16B16 RGB16	
	R16G16B16A16 RGBA16	

JPG Options

Command	Parameter	Description
-jpg_quality	<X>	Quality level (0..100). Save rate is X:1. Default is **75**.
-jpg_qualitysuperb		Sets the quality level to 100.
-jpg_qualitygood		Sets the quality level to 75.
-jpg_qualitynormal		Sets the quality level to 50.
-jpg_qualityaverage		Sets the quality level 25.
-jpg_qualitybad		Sets the quality level 10.
-jpg_progressive		Saves as a progressive JPEG file. Can be combined with the **-jpg_quality*** values.
-jpg_subsampling_411		Saves with high 4x1 chroma subsampling (4:1:1).
-jpg_subsampling_420		Saves with medium 2x2 chroma subsampling (4:2:0)—this is the defaul.
-jpg_subsampling_422		Saves with low 2x1 chroma subsampling (4:2:2).
-jpg_subsampling_444		Save with no chroma subsampling (4:4:4).
-jpg_optimize		Causes optional Huffman tables to be computed for the image. Can mildly reduce the file size.
-jpg_baseline		Saves as a basic JPEG file, without metadata or any markers.

Notes

Gamma

- Positive values use a raw pow(1/ γ)-based gamma curve. Negative values are divided into 2 halves: If γ is <= -1.0, an XtoLinear transform is applied; if -1.0 < γ < 0.0, a LinearToX transform is applied, where X is the curve specified by one of the standards (sRGB, ITU-R Recommendation BT.709-5, etc.) Generally, gamma will be above 1.0 or below -1.0.
- The default standard curve is sRGB Precise, so a default value of -2.2 results in a precise sRGB -> Linear transform.
- The relationship is reversed for target gamma. A positive value results in a pow(γ) transform being applied, while if γ is <= -1.0, a LinearToX transform is applied; if -1.0 < γ < 0.0, an XtoLinear transform is applied.
- This means that when considering gamma, you specify what the input is and what the target should be. By knowing what the input gamma is, the reverse transform can be applied to put the image back into linear space, and by knowing what the target gamma should be, a proper transform from linear to the target gamma can be made.

Texture Addressing

- Texture addressing is used during resampling and normal-map creation.
- For standard image resampling, **nul_border** is appropriate, as it will only sample from in-image texels, meaning no influence from outside 0..1 at all, and edge texels won't have an abnormally large influence as they would with **clamp**. For in-game textures, select the addressing mode that matches how it will be addressed in the game.

Cropping

- The addressing modes set with the `-textureaddressing` family of commands are shared with the cropping commands. `-crop` and `-crop3` use whatever addressing modes were set by the `-textureaddressing` family of commands, and the `-bake_tex_*` commands will overwrite any previous addressing modes set via the `-textureaddressing` commands and vice-versa.
- Each `-bake_tex_*` command overrides that axis of `-crop` or `-crop3`, even if `-crop` or `-crop3` comes after the `-bake_tex_*` command.

YUV

- By default, YUV <-> RGB algorithms that take additional parameters (Kr, Kb, Z, and B) are used. These provide reliable conversions, but many implementations use an approximation for these conversions, which can result in slight adjustments to the colors. Switching to the approximate conversion would be appropriate if you are dealing with a YUV file that has been adjusted to account for this color shift.
- The approximate YUV algorithm does not use Kr, Kb, Z, or B.
- YUV files that contain multiple frames are loaded as 3-D volume textures, with each slice of the depth component being a frame. You can resample the depth component to change the number of frames in the animation. This effectively smoothly speeds up or slows down the animation. Because this is a resampling through time rather than over color frequencies, a linear filter is most appropriate unless a specific visual effect is desired.

BMP

- Bit masks aren't used when saving a file as RLE.
- Some packed formats, such as A4R4G4B4, will only retain their component orders if bit masks are used. Without bit masks, the saved BMP file may have swizzled the components (for example to R4G4B4A4).

Formats

Below is a comprehensive list of formats to which and from which any loaded image can be converted. All formats can be supplied to the `–format` command, including the first word in the OpenGL format triplets (the internal format), however this will cause the first format encountered that matches the given OpenGL format to be selected, which may not be desired. For this reason, `-ogl_format` may be desired, which allows specifying the full OpenGL format.

Vulkan Formats

VK_FORMAT_R8_UNORM
VK_FORMAT_R8G8_UNORM
VK_FORMAT_R8G8B8_UNORM
VK_FORMAT_B8G8R8_UNORM
VK_FORMAT_R8G8B8A8_UNORM
VK_FORMAT_B8G8R8A8_UNORM
VK_FORMAT_R8_SNORM
VK_FORMAT_R8G8_SNORM

VK_FORMAT_R8G8B8_SNORM
VK_FORMAT_B8G8R8_SNORM
VK_FORMAT_R8G8B8A8_SNORM
VK_FORMAT_B8G8R8A8_SNORM
VK_FORMAT_R8_UINT
VK_FORMAT_R8G8_UINT
VK_FORMAT_R8G8B8_UINT
VK_FORMAT_B8G8R8_UINT
VK_FORMAT_R8G8B8A8_UINT
VK_FORMAT_B8G8R8A8_UINT
VK_FORMAT_R8_SINT
VK_FORMAT_R8G8_SINT
VK_FORMAT_R8G8B8_SINT
VK_FORMAT_B8G8R8_SINT
VK_FORMAT_R8G8B8A8_SINT
VK_FORMAT_B8G8R8A8_SINT
VK_FORMAT_R8_SRGB
VK_FORMAT_R8_SRGB
VK_FORMAT_R8G8_SRGB
VK_FORMAT_R8G8_SRGB
VK_FORMAT_R8G8B8_SRGB
VK_FORMAT_B8G8R8_SRGB
VK_FORMAT_R8G8B8A8_SRGB
VK_FORMAT_B8G8R8A8_SRGB
VK_FORMAT_R16_UNORM
VK_FORMAT_R16G16_UNORM
VK_FORMAT_R16G16B16_UNORM
VK_FORMAT_R16G16B16A16_UNORM
VK_FORMAT_R16_SNORM
VK_FORMAT_R16G16_SNORM
VK_FORMAT_R16G16B16_SNORM
VK_FORMAT_R16G16B16A16_SNORM
VK_FORMAT_R16_UINT
VK_FORMAT_R16G16_UINT
VK_FORMAT_R16G16B16_UINT
VK_FORMAT_R16G16B16A16_UINT
VK_FORMAT_R16_SINT
VK_FORMAT_R16G16_SINT
VK_FORMAT_R16G16B16_SINT
VK_FORMAT_R16G16B16A16_SINT
VK_FORMAT_R16_SFLOAT
VK_FORMAT_R16G16_SFLOAT
VK_FORMAT_R16G16B16_SFLOAT
VK_FORMAT_R16G16B16A16_SFLOAT
VK_FORMAT_R32_UINT

VK_FORMAT_R32G32_UINT
VK_FORMAT_R32G32B32_UINT
VK_FORMAT_R32G32B32A32_UINT
VK_FORMAT_R32_SINT
VK_FORMAT_R32G32_SINT
VK_FORMAT_R32G32B32_SINT
VK_FORMAT_R32G32B32A32_SINT
VK_FORMAT_R32_SFLOAT
VK_FORMAT_R32G32_SFLOAT
VK_FORMAT_R32G32B32_SFLOAT
VK_FORMAT_R32G32B32A32_SFLOAT
VK_FORMAT_R64_UINT
VK_FORMAT_R64G64_UINT
VK_FORMAT_R64G64B64_UINT
VK_FORMAT_R64G64B64A64_UINT
VK_FORMAT_R64_SINT
VK_FORMAT_R64G64_SINT
VK_FORMAT_R64G64B64_SINT
VK_FORMAT_R64G64B64A64_SINT
VK_FORMAT_R64_SFLOAT
VK_FORMAT_R64G64_SFLOAT
VK_FORMAT_R64G64B64_SFLOAT
VK_FORMAT_R64G64B64A64_SFLOAT
VK_FORMAT_R4G4_UNORM_PACK8
VK_FORMAT_R4G4B4A4_UNORM_PACK16
VK_FORMAT_B4G4R4A4_UNORM_PACK16
VK_FORMAT_A4R4G4B4_UNORM_PACK16
VK_FORMAT_A4B4G4R4_UNORM_PACK16
VK_FORMAT_R5G6B5_UNORM_PACK16
VK_FORMAT_B5G6R5_UNORM_PACK16
VK_FORMAT_R5G5B5A1_UNORM_PACK16
VK_FORMAT_A1B5G5R5_UNORM_PACK16_KHR
VK_FORMAT_A1R5G5B5_UNORM_PACK16
VK_FORMAT_A8B8G8R8_UNORM_PACK32
VK_FORMAT_A8B8G8R8_SNORM_PACK32
VK_FORMAT_A8B8G8R8_UINT_PACK32
VK_FORMAT_A8B8G8R8_SINT_PACK32
VK_FORMAT_A8B8G8R8_SRGB_PACK32
VK_FORMAT_A2R10G10B10_UNORM_PACK32
VK_FORMAT_A2B10G10R10_UNORM_PACK32
VK_FORMAT_A2R10G10B10_SNORM_PACK32
VK_FORMAT_A2B10G10R10_SNORM_PACK32
VK_FORMAT_A2R10G10B10_UINT_PACK32
VK_FORMAT_A2B10G10R10_UINT_PACK32
VK_FORMAT_A2R10G10B10_SINT_PACK32

VK_FORMAT_A2B10G10R10_SINT_PACK32
VK_FORMAT_B10G11R11_UFLOAT_PACK32
VK_FORMAT_E5B9G9R9_UFLOAT_PACK32
VK_FORMAT_A8_UNORM_KHR
VK_FORMAT_BC1_RGB_UNORM_BLOCK
VK_FORMAT_BC1_RGBA_UNORM_BLOCK
VK_FORMAT_BC2_UNORM_BLOCK
VK_FORMAT_BC3_UNORM_BLOCK
VK_FORMAT_BC1_RGB_SRGB_BLOCK
VK_FORMAT_BC1_RGBA_SRGB_BLOCK
VK_FORMAT_BC2_SRGB_BLOCK
VK_FORMAT_BC3_SRGB_BLOCK
VK_FORMAT_BC4_UNORM_BLOCK
VK_FORMAT_BC5_UNORM_BLOCK
VK_FORMAT_BC4_SNORM_BLOCK
VK_FORMAT_BC5_SNORM_BLOCK
VK_FORMAT_BC6H_UFLOAT_BLOCK
VK_FORMAT_BC7_UNORM_BLOCK
VK_FORMAT_BC7_SRGB_BLOCK
VK_FORMAT_ETC2_R8G8B8_UNORM_BLOCK
VK_FORMAT_ETC2_R8G8B8A1_UNORM_BLOCK
VK_FORMAT_ETC2_R8G8B8A8_UNORM_BLOCK
VK_FORMAT_ETC2_R8G8B8_SRGB_BLOCK
VK_FORMAT_ETC2_R8G8B8A1_SRGB_BLOCK
VK_FORMAT_ETC2_R8G8B8A8_SRGB_BLOCK
VK_FORMAT_EAC_R11_UNORM_BLOCK
VK_FORMAT_EAC_R11G11_UNORM_BLOCK
VK_FORMAT_EAC_R11_SNORM_BLOCK
VK_FORMAT_EAC_R11G11_SNORM_BLOCK
VK_FORMAT_PVRTC1_2BPP_UNORM_BLOCK_IMG
VK_FORMAT_PVRTC1_4BPP_UNORM_BLOCK_IMG
VK_FORMAT_PVRTC2_2BPP_UNORM_BLOCK_IMG
VK_FORMAT_PVRTC2_4BPP_UNORM_BLOCK_IMG
VK_FORMAT_PVRTC1_2BPP_SRGB_BLOCK_IMG
VK_FORMAT_PVRTC1_4BPP_SRGB_BLOCK_IMG
VK_FORMAT_PVRTC2_2BPP_SRGB_BLOCK_IMG
VK_FORMAT_PVRTC2_4BPP_SRGB_BLOCK_IMG
VK_FORMAT_ASTC_4x4_UNORM_BLOCK
VK_FORMAT_ASTC_5x4_UNORM_BLOCK
VK_FORMAT_ASTC_5x5_UNORM_BLOCK
VK_FORMAT_ASTC_6x5_UNORM_BLOCK
VK_FORMAT_ASTC_6x6_UNORM_BLOCK
VK_FORMAT_ASTC_8x5_UNORM_BLOCK
VK_FORMAT_ASTC_8x6_UNORM_BLOCK
VK_FORMAT_ASTC_8x8_UNORM_BLOCK

VK_FORMAT_ASTC_10x5_UNORM_BLOCK
VK_FORMAT_ASTC_10x6_UNORM_BLOCK
VK_FORMAT_ASTC_10x8_UNORM_BLOCK
VK_FORMAT_ASTC_10x10_UNORM_BLOCK
VK_FORMAT_ASTC_12x10_UNORM_BLOCK
VK_FORMAT_ASTC_12x12_UNORM_BLOCK
VK_FORMAT_ASTC_4x4_SRGB_BLOCK
VK_FORMAT_ASTC_5x4_SRGB_BLOCK
VK_FORMAT_ASTC_5x5_SRGB_BLOCK
VK_FORMAT_ASTC_6x5_SRGB_BLOCK
VK_FORMAT_ASTC_6x6_SRGB_BLOCK
VK_FORMAT_ASTC_8x5_SRGB_BLOCK
VK_FORMAT_ASTC_8x6_SRGB_BLOCK
VK_FORMAT_ASTC_8x8_SRGB_BLOCK
VK_FORMAT_ASTC_10x5_SRGB_BLOCK
VK_FORMAT_ASTC_10x6_SRGB_BLOCK
VK_FORMAT_ASTC_10x8_SRGB_BLOCK
VK_FORMAT_ASTC_10x10_SRGB_BLOCK
VK_FORMAT_ASTC_12x10_SRGB_BLOCK
VK_FORMAT_ASTC_12x12_SRGB_BLOCK
VK_FORMAT_ASTC_4x4_SFLOAT_BLOCK
VK_FORMAT_ASTC_5x4_SFLOAT_BLOCK
VK_FORMAT_ASTC_5x5_SFLOAT_BLOCK
VK_FORMAT_ASTC_6x5_SFLOAT_BLOCK
VK_FORMAT_ASTC_6x6_SFLOAT_BLOCK
VK_FORMAT_ASTC_8x5_SFLOAT_BLOCK
VK_FORMAT_ASTC_8x6_SFLOAT_BLOCK
VK_FORMAT_ASTC_8x8_SFLOAT_BLOCK
VK_FORMAT_ASTC_10x5_SFLOAT_BLOCK
VK_FORMAT_ASTC_10x6_SFLOAT_BLOCK
VK_FORMAT_ASTC_10x8_SFLOAT_BLOCK
VK_FORMAT_ASTC_10x10_SFLOAT_BLOCK
VK_FORMAT_ASTC_12x10_SFLOAT_BLOCK
VK_FORMAT_ASTC_12x12_SFLOAT_BLOCK
VK_FORMAT_ASTC_3x3x3_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_4x3x3_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_4x4x3_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_4x4x4_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_5x4x4_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_5x5x4_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_5x5x5_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_6x5x5_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_6x6x5_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_6x6x6_UNORM_BLOCK_EXT
VK_FORMAT_ASTC_3x3x3_SRGB_BLOCK_EXT

VK_FORMAT_ASTC_4x3x3_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_4x4x3_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_4x4x4_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_5x4x4_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_5x5x4_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_5x5x5_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_6x5x5_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_6x6x5_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_6x6x6_SRGB_BLOCK_EXT
VK_FORMAT_ASTC_3x3x3_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_4x3x3_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_4x4x3_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_4x4x4_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_5x4x4_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_5x5x4_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_5x5x5_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_6x5x5_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_6x6x5_SFLOAT_BLOCK_EXT
VK_FORMAT_ASTC_6x6x6_SFLOAT_BLOCK_EXT
VK_FORMAT_D16_UNORM
VK_FORMAT_D32_SFLOAT
VK_FORMAT_S8_UINT
VK_FORMAT_D16_UNORM_S8_UINT
VK_FORMAT_D24_UNORM_S8_UINT
VK_FORMAT_X8_D24_UNORM_PACK32
VK_FORMAT_D32_SFLOAT_S8_UINT
VK_FORMAT_G16_B16_R16_3PLANE_444_UNORM
VK_FORMAT_G12X4_B12X4_R12X4_3PLANE_444_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6_R10X6_3PLANE_444_UNORM_3PACK16
VK_FORMAT_G8_B8_R8_3PLANE_444_UNORM
VK_FORMAT_G16_B16R16_2PLANE_444_UNORM
VK_FORMAT_G12X4_B12X4R12X4_2PLANE_444_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6R10X6_2PLANE_444_UNORM_3PACK16
VK_FORMAT_G8_B8R8_2PLANE_444_UNORM
VK_FORMAT_G16_B16_R16_3PLANE_422_UNORM
VK_FORMAT_G12X4_B12X4_R12X4_3PLANE_422_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6_R10X6_3PLANE_422_UNORM_3PACK16
VK_FORMAT_G8_B8_R8_3PLANE_422_UNORM
VK_FORMAT_G16_B16R16_2PLANE_422_UNORM
VK_FORMAT_G12X4_B12X4R12X4_2PLANE_422_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6R10X6_2PLANE_422_UNORM_3PACK16
VK_FORMAT_G8_B8R8_2PLANE_422_UNORM
VK_FORMAT_G16_B16_R16_3PLANE_420_UNORM
VK_FORMAT_G12X4_B12X4_R12X4_3PLANE_420_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6_R10X6_3PLANE_420_UNORM_3PACK16

VK_FORMAT_G8_B8_R8_3PLANE_420_UNORM
VK_FORMAT_G16_B16R16_2PLANE_420_UNORM
VK_FORMAT_G12X4_B12X4R12X4_2PLANE_420_UNORM_3PACK16
VK_FORMAT_G10X6_B10X6R10X6_2PLANE_420_UNORM_3PACK16
VK_FORMAT_G8_B8R8_2PLANE_420_UNORM
VK_FORMAT_G16B16G16R16_422_UNORM
VK_FORMAT_G12X4B12X4G12X4R12X4_422_UNORM_4PACK16
VK_FORMAT_G10X6B10X6G10X6R10X6_422_UNORM_4PACK16
VK_FORMAT_G8B8G8R8_422_UNORM
VK_FORMAT_B16G16R16G16_422_UNORM
VK_FORMAT_B12X4G12X4R12X4G12X4_422_UNORM_4PACK16
VK_FORMAT_B10X6G10X6R10X6G10X6_422_UNORM_4PACK16
VK_FORMAT_B8G8R8G8_422_UNORM
VK_FORMAT_R12X4G12X4B12X4A12X4_UNORM_4PACK16
VK_FORMAT_R10X6G10X6B10X6A10X6_UNORM_4PACK16

DXGI Formats

DXGI_FORMAT_R8_UNORM
DXGI_FORMAT_R8G8_UNORM
DXGI_FORMAT_R8G8B8A8_UNORM
DXGI_FORMAT_B8G8R8A8_UNORM
DXGI_FORMAT_B8G8R8X8_UNORM
DXGI_FORMAT_R8_SNORM
DXGI_FORMAT_R8G8_SNORM
DXGI_FORMAT_R8G8B8A8_SNORM
DXGI_FORMAT_R8_UINT
DXGI_FORMAT_R8G8_UINT
DXGI_FORMAT_R8G8B8A8_UINT
DXGI_FORMAT_R8_SINT
DXGI_FORMAT_R8G8_SINT
DXGI_FORMAT_R8G8B8A8_SINT
DXGI_FORMAT_R8G8B8A8_UNORM_SRGB
DXGI_FORMAT_B8G8R8A8_UNORM_SRGB
DXGI_FORMAT_B8G8R8X8_UNORM_SRGB
DXGI_FORMAT_R16_UNORM
DXGI_FORMAT_R16G16_UNORM
DXGI_FORMAT_R16G16B16A16_UNORM
DXGI_FORMAT_R16_SNORM
DXGI_FORMAT_R16G16_SNORM
DXGI_FORMAT_R16G16B16A16_SNORM
DXGI_FORMAT_R16_UINT
DXGI_FORMAT_R16G16_UINT
DXGI_FORMAT_R16G16B16A16_UINT
DXGI_FORMAT_R16_SINT
DXGI_FORMAT_R16G16_SINT

DXGI_FORMAT_R16G16B16A16_SINT
DXGI_FORMAT_R16_FLOAT
DXGI_FORMAT_R16G16_FLOAT
DXGI_FORMAT_R16G16B16A16_FLOAT
DXGI_FORMAT_R32_UINT
DXGI_FORMAT_R32G32_UINT
DXGI_FORMAT_R32G32B32_UINT
DXGI_FORMAT_R32G32B32A32_UINT
DXGI_FORMAT_R32_SINT
DXGI_FORMAT_R32G32_SINT
DXGI_FORMAT_R32G32B32_SINT
DXGI_FORMAT_R32G32B32A32_SINT
DXGI_FORMAT_R32_FLOAT
DXGI_FORMAT_R32G32_FLOAT
DXGI_FORMAT_R32G32B32_FLOAT
DXGI_FORMAT_R32G32B32A32_FLOAT
DXGI_FORMAT_B4G4R4A4_UNORM
DXGI_FORMAT_B5G6R5_UNORM
DXGI_FORMAT_B5G5R5A1_UNORM
DXGI_FORMAT_R10G10B10_XR_BIAS_A2_UNORM
DXGI_FORMAT_R10G10B10A2_UNORM
DXGI_FORMAT_R10G10B10A2_UINT
DXGI_FORMAT_R11G11B10_FLOAT
DXGI_FORMAT_R9G9B9E5_SHAREDEXP
DXGI_FORMAT_A8_UNORM
DXGI_FORMAT_BC1_UNORM
DXGI_FORMAT_BC2_UNORM
DXGI_FORMAT_BC3_UNORM
DXGI_FORMAT_BC1_UNORM_SRGB
DXGI_FORMAT_BC2_UNORM_SRGB
DXGI_FORMAT_BC3_UNORM_SRGB
DXGI_FORMAT_BC4_UNORM
DXGI_FORMAT_BC5_UNORM
DXGI_FORMAT_BC4_SNORM
DXGI_FORMAT_BC5_SNORM
DXGI_FORMAT_BC6H_UF16
DXGI_FORMAT_BC7_UNORM
DXGI_FORMAT_BC7_UNORM_SRGB
DXGI_FORMAT_D16_UNORM
DXGI_FORMAT_D32_FLOAT
DXGI_FORMAT_D24_UNORM_S8_UINT
DXGI_FORMAT_D32_FLOAT_S8X24_UINT
DXGI_FORMAT_P216
DXGI_FORMAT_P210
DXGI_FORMAT_P208

DXGI_FORMAT_420_OPAQUE
DXGI_FORMAT_YV12
DXGI_FORMAT_P016
DXGI_FORMAT_P010
DXGI_FORMAT_NV12
DXGI_FORMAT_NV21
DXGI_FORMAT_Y216
DXGI_FORMAT_Y210
DXGI_FORMAT_G8R8_G8B8_UNORM
DXGI_FORMAT_YUY2
DXGI_FORMAT_R8G8_B8G8_UNORM
DXGI_FORMAT_Y416
DXGI_FORMAT_Y410
DXGI_FORMAT_AYUV

Metal Formats

MTLPixelFormatR8Unorm
MTLPixelFormatRG8Unorm
MTLPixelFormatRGBA8Unorm
MTLPixelFormatBGRA8Unorm
MTLPixelFormatR8Snorm
MTLPixelFormatRG8Snorm
MTLPixelFormatRGBA8Snorm
MTLPixelFormatR8Uint
MTLPixelFormatRG8Uint
MTLPixelFormatRGBA8Uint
MTLPixelFormatR8Sint
MTLPixelFormatRG8Sint
MTLPixelFormatRGBA8Sint
MTLPixelFormatR8Unorm_sRGB
MTLPixelFormatR8Unorm_sRGB
MTLPixelFormatRG8Unorm_sRGB
MTLPixelFormatRG8Unorm_sRGB
MTLPixelFormatRGBA8Unorm_sRGB
MTLPixelFormatBGRA8Unorm_sRGB
MTLPixelFormatR16Unorm
MTLPixelFormatRG16Unorm
MTLPixelFormatRGBA16Unorm
MTLPixelFormatR16Snorm
MTLPixelFormatRG16Snorm
MTLPixelFormatRGBA16Snorm
MTLPixelFormatR16Uint
MTLPixelFormatRG16Uint
MTLPixelFormatRGBA16Uint
MTLPixelFormatR16Sint

MTLPixelFormatRG16Sint
MTLPixelFormatRGBA16Sint
MTLPixelFormatR16Float
MTLPixelFormatRG16Float
MTLPixelFormatRGBA16Float
MTLPixelFormatR32Uint
MTLPixelFormatRG32Uint
MTLPixelFormatRGBA32Uint
MTLPixelFormatR32Sint
MTLPixelFormatRG32Sint
MTLPixelFormatRGBA32Sint
MTLPixelFormatR32Float
MTLPixelFormatRG32Float
MTLPixelFormatRGBA32Float
MTLPixelFormatABGR4Unorm
MTLPixelFormatB5G6R5Unorm
MTLPixelFormatA1BGR5Unorm
MTLPixelFormatBGR5A1Unorm
MTLPixelFormatBGR10A2Unorm
MTLPixelFormatRGB10A2Unorm
MTLPixelFormatRGB10A2Uint
MTLPixelFormatRG11B10Float
MTLPixelFormatRGB9E5Float
MTLPixelFormatA8Unorm
MTLPixelFormatBC1_RGBA
MTLPixelFormatBC2_RGBA
MTLPixelFormatBC3_RGBA
MTLPixelFormatBC1_RGBA_sRGB
MTLPixelFormatBC2_RGBA_sRGB
MTLPixelFormatBC3_RGBA_sRGB
MTLPixelFormatBC4_RUnorm
MTLPixelFormatBC5_RGUnorm
MTLPixelFormatBC4_RSnorm
MTLPixelFormatBC5_RGSnorm
MTLPixelFormatBC6H_RGBUfloat
MTLPixelFormatBC7_RGBAUnorm
MTLPixelFormatBC7_RGBAUnorm_sRGB
MTLPixelFormatETC2_RGB8
MTLPixelFormatETC2_RGB8A1
MTLPixelFormatEAC_RGBA8
MTLPixelFormatETC2_RGB8_sRGB
MTLPixelFormatETC2_RGB8A1_sRGB
MTLPixelFormatEAC_RGBA8_sRGB
MTLPixelFormatEAC_R11Unorm
MTLPixelFormatEAC_RG11Unorm

MTLPixelFormatEAC_R11Snorm
MTLPixelFormatEAC_RG11Snorm
MTLPixelFormatPVRTC_RGB_2BPP
MTLPixelFormatPVRTC_RGB_4BPP
MTLPixelFormatPVRTC_RGBA_2BPP
MTLPixelFormatPVRTC_RGBA_4BPP
MTLPixelFormatPVRTC_RGB_2BPP_sRGB
MTLPixelFormatPVRTC_RGB_4BPP_sRGB
MTLPixelFormatPVRTC_RGBA_2BPP_sRGB
MTLPixelFormatPVRTC_RGBA_4BPP_sRGB
MTLPixelFormatASTC_4x4_LDR
MTLPixelFormatASTC_5x4_LDR
MTLPixelFormatASTC_5x5_LDR
MTLPixelFormatASTC_6x5_LDR
MTLPixelFormatASTC_6x6_LDR
MTLPixelFormatASTC_8x5_LDR
MTLPixelFormatASTC_8x6_LDR
MTLPixelFormatASTC_8x8_LDR
MTLPixelFormatASTC_10x5_LDR
MTLPixelFormatASTC_10x6_LDR
MTLPixelFormatASTC_10x8_LDR
MTLPixelFormatASTC_10x10_LDR
MTLPixelFormatASTC_12x10_LDR
MTLPixelFormatASTC_12x12_LDR
MTLPixelFormatASTC_4x4_sRGB
MTLPixelFormatASTC_5x4_sRGB
MTLPixelFormatASTC_5x5_sRGB
MTLPixelFormatASTC_6x5_sRGB
MTLPixelFormatASTC_6x6_sRGB
MTLPixelFormatASTC_8x5_sRGB
MTLPixelFormatASTC_8x6_sRGB
MTLPixelFormatASTC_8x8_sRGB
MTLPixelFormatASTC_10x5_sRGB
MTLPixelFormatASTC_10x6_sRGB
MTLPixelFormatASTC_10x8_sRGB
MTLPixelFormatASTC_10x10_sRGB
MTLPixelFormatASTC_12x10_sRGB
MTLPixelFormatASTC_12x12_sRGB
MTLPixelFormatASTC_4x4_HDR
MTLPixelFormatASTC_5x4_HDR
MTLPixelFormatASTC_5x5_HDR
MTLPixelFormatASTC_6x5_HDR
MTLPixelFormatASTC_6x6_HDR
MTLPixelFormatASTC_8x5_HDR
MTLPixelFormatASTC_8x6_HDR

MTLPixelFormatASTC_8x8_HDR
MTLPixelFormatASTC_10x5_HDR
MTLPixelFormatASTC_10x6_HDR
MTLPixelFormatASTC_10x8_HDR
MTLPixelFormatASTC_10x10_HDR
MTLPixelFormatASTC_12x10_HDR
MTLPixelFormatASTC_12x12_HDR
MTLPixelFormatDepth16Unorm
MTLPixelFormatDepth32Float
MTLPixelFormatStencil8
MTLPixelFormatDepth24Unorm_Stencil8
MTLPixelFormatDepth32Float_Stencil8
MTLPixelFormatGBGR422
MTLPixelFormatBGRG422

OpenGL Formats

glInternalFormat	glType	glBaseInternalFormat
GL_R8	GL_UNSIGNED_BYTE	GL_RED
GL_RG8	GL_UNSIGNED_BYTE	GL_RG
GL_RGB8	GL_UNSIGNED_BYTE	GL_RGB
GL_RGB8	GL_UNSIGNED_BYTE	GL_BGR
GL_RGBA8	GL_UNSIGNED_BYTE	GL_RGBA
GL_RGBA8	GL_UNSIGNED_BYTE	GL_BGRA
GL_R8_SNORM	GL_BYTE	GL_RED
GL_RG8_SNORM	GL_BYTE	GL_RG
GL_RGB8_SNORM	GL_BYTE	GL_RGB
GL_RGB8_SNORM	GL_BYTE	GL_BGR
GL_RGBA8_SNORM	GL_BYTE	GL_RGBA
GL_RGBA8_SNORM	GL_BYTE	GL_BGRA
GL_R8UI	GL_UNSIGNED_BYTE	GL_RED_INTEGER
GL_RG8UI	GL_UNSIGNED_BYTE	GL_RG_INTEGER
GL_RGB8UI	GL_UNSIGNED_BYTE	GL_RGB_INTEGER
GL_RGB8UI	GL_UNSIGNED_BYTE	GL_BGR_INTEGER
GL_RGBA8UI	GL_UNSIGNED_BYTE	GL_RGBA_INTEGER
GL_RGBA8UI	GL_UNSIGNED_BYTE	GL_BGRA_INTEGER
GL_R8I	GL_BYTE	GL_RED_INTEGER
GL_RG8I	GL_BYTE	GL_RG_INTEGER
GL_RGB8I	GL_BYTE	GL_RGB_INTEGER
GL_RGB8I	GL_BYTE	GL_BGR_INTEGER
GL_RGBA8I	GL_BYTE	GL_RGBA_INTEGER

GL_RGBA8I	GL_BYTE	GL_BGRA_INTEGER
GL_SR8	GL_UNSIGNED_BYTE	GL_RED
GL_SR8_EXT	GL_UNSIGNED_BYTE	GL_RED
GL_SRG8	GL_UNSIGNED_BYTE	GL_RG
GL_SRG8_EXT	GL_UNSIGNED_BYTE	GL_RG
GL_SRGB8	GL_UNSIGNED_BYTE	GL_RGB
GL_SRGB8	GL_UNSIGNED_BYTE	GL_BGR
GL_SRGB8_ALPHA8	GL_UNSIGNED_BYTE	GL_RGBA
GL_SRGB8_ALPHA8	GL_UNSIGNED_BYTE	GL_BGRA
GL_R16	GL_UNSIGNED_SHORT	GL_RED
GL_RG16	GL_UNSIGNED_SHORT	GL_RG
GL_RGB16	GL_UNSIGNED_SHORT	GL_RGB
GL_RGBA16	GL_UNSIGNED_SHORT	GL_RGBA
GL_R16_SNORM	GL_SHORT	GL_RED
GL_RG16_SNORM	GL_SHORT	GL_RG
GL_RGB16_SNORM	GL_SHORT	GL_RGB
GL_RGBA16_SNORM	GL_SHORT	GL_RGBA
GL_R16UI	GL_UNSIGNED_SHORT	GL_RED_INTEGER
GL_RG16UI	GL_UNSIGNED_SHORT	GL_RG_INTEGER
GL_RGB16UI	GL_UNSIGNED_SHORT	GL_RGB_INTEGER
GL_RGBA16UI	GL_UNSIGNED_SHORT	GL_RGBA_INTEGER
GL_R16I	GL_SHORT	GL_RED_INTEGER
GL_RG16I	GL_SHORT	GL_RG_INTEGER
GL_RGB16I	GL_SHORT	GL_RGB_INTEGER
GL_RGBA16I	GL_SHORT	GL_RGBA_INTEGER
GL_R16F	GL_HALF_FLOAT	GL_RED
GL_RG16F	GL_HALF_FLOAT	GL_RG
GL_RGB16F	GL_HALF_FLOAT	GL_RGB
GL_RGBA16F	GL_HALF_FLOAT	GL_RGBA
GL_R32UI	GL_UNSIGNED_INT	GL_RED_INTEGER
GL_RG32UI	GL_UNSIGNED_INT	GL_RG_INTEGER
GL_RGB32UI	GL_UNSIGNED_INT	GL_RGB_INTEGER
GL_RGBA32UI	GL_UNSIGNED_INT	GL_RGBA_INTEGER
GL_R32I	GL_INT	GL_RED_INTEGER
GL_RG32I	GL_INT	GL_RG_INTEGER
GL_RGB32I	GL_INT	GL_RGB_INTEGER
GL_RGBA32I	GL_INT	GL_RGBA_INTEGER
GL_R32F	GL_FLOAT	GL_RED
GL_RG32F	GL_FLOAT	GL_RG
GL_RGB32F	GL_FLOAT	GL_RGB
GL_RGBA32F	GL_FLOAT	GL_RGBA

GL_R3_G3_B2	GL_UNSIGNED_BYTE_2_3_3_REV	GL_RGB
GL_RGB4	GL_UNSIGNED_SHORT_4_4_4_4	GL_RGB
GL_RGB4	GL_UNSIGNED_SHORT_4_4_4_4_REV	GL_RGB
GL_RGBA4	GL_UNSIGNED_SHORT_4_4_4_4	GL_RGBA
GL_RGBA4	GL_UNSIGNED_SHORT_4_4_4_4	GL_BGRA
GL_RGBA4	GL_UNSIGNED_SHORT_4_4_4_4_REV	GL_BGRA
GL_RGBA4	GL_UNSIGNED_SHORT_4_4_4_4_REV	GL_RGBA
GL_RGB5	GL_UNSIGNED_SHORT_1_5_5_5_REV	GL_RGB
GL_RGB565	GL_UNSIGNED_SHORT_5_6_5	GL_RGB
GL_RGB565	GL_UNSIGNED_SHORT_5_6_5_REV	GL_RGB
GL_RGB10	GL_UNSIGNED_SHORT	GL_RGB
GL_RGB12	GL_UNSIGNED_SHORT	GL_RGB
GL_RGBA2	GL_UNSIGNED_BYTE	GL_RGBA
GL_RGBA12	GL_UNSIGNED_SHORT	GL_RGBA
GL_RGB5_A1	GL_UNSIGNED_SHORT_5_5_5_1	GL_RGBA
GL_RGB5_A1	GL_UNSIGNED_SHORT_5_5_5_1	GL_BGRA
GL_RGB5_A1	GL_UNSIGNED_SHORT_1_5_5_5_REV	GL_RGBA
GL_RGB5_A1	GL_UNSIGNED_SHORT_1_5_5_5_REV	GL_BGRA
GL_RGBA8	GL_UNSIGNED_BYTE_3_3_2	GL_BGRA
GL_RGB10_A2	GL_UNSIGNED_INT_2_10_10_10_REV	GL_BGRA
GL_RGB10_A2	GL_UNSIGNED_INT_2_10_10_10_REV	GL_RGBA
GL_RGB10_A2UI	GL_UNSIGNED_INT_2_10_10_10_REV	GL_BGRA_INTEGER
GL_RGB10_A2UI	GL_UNSIGNED_INT_2_10_10_10_REV	GL_RGBA_INTEGER
GL_R11F_G11F_B10F	GL_UNSIGNED_INT_10F_11F_11F_REV	GL_RGB
GL_RGB9_E5	GL_UNSIGNED_INT_5_9_9_9_REV	GL_RGB

GL_ALPHA4	GL_UNSIGNED_BYTE	GL_ALPHA
GL_ALPHA8	GL_UNSIGNED_BYTE	GL_ALPHA
GL_ALPHA8_SNORM	GL_BYTE	GL_ALPHA
GL_ALPHA8UI_EXT	GL_UNSIGNED_BYTE	GL_ALPHA_INTEGER
GL_ALPHA8I_EXT	GL_BYTE	GL_ALPHA_INTEGER
GL_ALPHA12	GL_UNSIGNED_SHORT	GL_ALPHA
GL_ALPHA16	GL_UNSIGNED_SHORT	GL_ALPHA
GL_ALPHA16_SNORM	GL_SHORT	GL_ALPHA
GL_ALPHA16UI_EXT	GL_UNSIGNED_SHORT	GL_ALPHA_INTEGER
GL_ALPHA16I_EXT	GL_SHORT	GL_ALPHA_INTEGER
GL_ALPHA16F_ARB	GL_HALF_FLOAT	GL_ALPHA
GL_ALPHA32UI_EXT	GL_UNSIGNED_INT	GL_ALPHA_INTEGER
GL_ALPHA32I_EXT	GL_INT	GL_ALPHA_INTEGER
GL_ALPHA32F_ARB	GL_FLOAT	GL_ALPHA
GL_LUMINANCE4	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_LUMINANCE8	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_LUMINANCE8_SNORM	GL_BYTE	GL_LUMINANCE
GL_SLUMINANCE8	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_LUMINANCE8UI_EXT	GL_UNSIGNED_BYTE	GL_LUMINANCE_INTEGER
GL_LUMINANCE8I_EXT	GL_BYTE	GL_LUMINANCE_INTEGER
GL_LUMINANCE12	GL_UNSIGNED_SHORT	GL_LUMINANCE
GL_LUMINANCE16	GL_UNSIGNED_SHORT	GL_LUMINANCE
GL_LUMINANCE16_SNORM	GL_SHORT	GL_LUMINANCE
GL_LUMINANCE16UI_EXT	GL_UNSIGNED_SHORT	GL_LUMINANCE_INTEGER
GL_LUMINANCE16I_EXT	GL_SHORT	GL_LUMINANCE_INTEGER
GL_LUMINANCE16F_ARB	GL_HALF_FLOAT	GL_LUMINANCE
GL_LUMINANCE32UI_EXT	GL_UNSIGNED_INT	GL_LUMINANCE_INTEGER
GL_LUMINANCE32I_EXT	GL_INT	GL_LUMINANCE_INTEGER
GL_LUMINANCE32F_ARB	GL_FLOAT	GL_LUMINANCE
GL_LUMINANCE4_ALPHA4	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA
GL_LUMINANCE6_ALPHA2	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA

GL_LUMINANCE8_ALPHA8	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA
GL_LUMINANCE8_ALPHA8_SNORM	GL_BYTE	GL_LUMINANCE_ALPHA
GL_SLUMINANCE8_ALPHA8	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA
GL_LUMINANCE_ALPHA8UI_EXT	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE_ALPHA8I_EXT	GL_BYTE	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE12_ALPHA4	GL_UNSIGNED_SHORT	GL_LUMINANCE_ALPHA
GL_LUMINANCE12_ALPHA12	GL_UNSIGNED_SHORT	GL_LUMINANCE_ALPHA
GL_LUMINANCE16_ALPHA16	GL_UNSIGNED_SHORT	GL_LUMINANCE_ALPHA
GL_LUMINANCE16_ALPHA16_SNORM	GL_SHORT	GL_LUMINANCE_ALPHA
GL_LUMINANCE_ALPHA16UI_EXT	GL_UNSIGNED_SHORT	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE_ALPHA16I_EXT	GL_SHORT	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE_ALPHA16F_ARB	GL_HALF_FLOAT	GL_LUMINANCE_ALPHA
GL_LUMINANCE_ALPHA32UI_EXT	GL_UNSIGNED_INT	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE_ALPHA32I_EXT	GL_INT	GL_LUMINANCE_ALPHA_INTEGER
GL_LUMINANCE_ALPHA32F_ARB	GL_FLOAT	GL_LUMINANCE_ALPHA
GL_INTENSITY4	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_INTENSITY8	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_INTENSITY8_SNORM	GL_BYTE	GL_LUMINANCE
GL_INTENSITY8UI_EXT	GL_UNSIGNED_BYTE	GL_LUMINANCE_INTEGER
GL_INTENSITY8I_EXT	GL_BYTE	GL_LUMINANCE_INTEGER
GL_INTENSITY12	GL_UNSIGNED_SHORT	GL_LUMINANCE
GL_INTENSITY16	GL_UNSIGNED_SHORT	GL_LUMINANCE
GL_INTENSITY16_SNORM	GL_SHORT	GL_LUMINANCE
GL_INTENSITY16UI_EXT	GL_UNSIGNED_SHORT	GL_LUMINANCE_INTEGER
GL_INTENSITY16I_EXT	GL_SHORT	GL_LUMINANCE_INTEGER
GL_INTENSITY16F_ARB	GL_HALF_FLOAT	GL_LUMINANCE

GL_INTENSITY32UI_EXT	GL_UNSIGNED_INT	GL_LUMINANCE_INTEGER
GL_INTENSITY32I_EXT	GL_INT	GL_LUMINANCE_INTEGER
GL_INTENSITY32F_ARB	GL_FLOAT	GL_LUMINANCE
GL_COMPRESSED_RED	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_ALPHA	GL_UNSIGNED_BYTE	GL_ALPHA
GL_COMPRESSED_LUMINANCE	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_COMPRESSED_SLUMINANCE	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_COMPRESSED_LUMINANCE_ALPHA	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA
GL_COMPRESSED_SLUMINANCE_ALPHA	GL_UNSIGNED_BYTE	GL_LUMINANCE_ALPHA
GL_COMPRESSED_INTENSITY	GL_UNSIGNED_BYTE	GL_LUMINANCE
GL_COMPRESSED_RG	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_RGB	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGBA	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_SRGB_ALPHA	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGB_S3TC_DXT1_EXT	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGBA_S3TC_DXT1_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_S3TC_DXT3_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_S3TC_DXT5_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_S3TC_DXT1_EXT	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_SRGB_ALPHA_S3TC_DXT1_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_ALPHA_S3TC_DXT3_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_ALPHA_S3TC_DXT5_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_LUMINANCE_LATC1_EXT	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_LUMINANCE_ALPHA_LATC2_EXT	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_SIGNED_LUMINANCE_LATC1_EXT	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_SIGNED_LUMINANCE_ALPHA_LATC2_EXT	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_RED_RGTC1	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_RG_RGTC2	GL_UNSIGNED_BYTE	GL_RG

GL_COMPRESSED_SIGNED_RED_RGTC1	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_SIGNED_RG_RGTC2	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT	GL_FLOAT	GL_RGB
GL_COMPRESSED_RGBA_BPTC_UNORM	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM	GL_UNSIGNED_BYTE	GL_RGBA
GL_ETC1_RGB8_OES	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGB8_ETC2	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGB8_PUNCHTHROUGH_ALPHA1_ETC2	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA8_ETC2_EAC	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ETC2	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_SRGB8_PUNCHTHROUGH_ALPHA1_ETC2	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ETC2_EAC	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_R11_EAC	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_RG11_EAC	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_SIGNED_R11_EAC	GL_UNSIGNED_BYTE	GL_RED
GL_COMPRESSED_SIGNED_RG11_EAC	GL_UNSIGNED_BYTE	GL_RG
GL_COMPRESSED_RGB_PVRTC_2BPPV1_IMG	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGB_PVRTC_4BPPV1_IMG	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_RGBA_PVRTC_2BPPV1_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_PVRTC_4BPPV1_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_PVRTC_2BPPV2_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_PVRTC_4BPPV2_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_PVRTC_2BPPV1_EXT	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_SRGB_PVRTC_4BPPV1_EXT	GL_UNSIGNED_BYTE	GL_RGB
GL_COMPRESSED_SRGB_ALPHA_PVRTC_2BPPV1_EXT	GL_UNSIGNED_BYTE	GL_RGBA

GL_COMPRESSED_SRGB_ALPHA_PVRTC_4BPPV1_EXT	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_ALPHA_PVRTC_2BPPV2_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB_ALPHA_PVRTC_4BPPV2_IMG	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_4x4_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_5x4_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_5x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_6x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_6x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_8x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_8x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_8x8_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_10x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_10x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_10x8_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_10x10_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_12x10_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_12x12_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_4x4_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_5x4_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_5x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_6x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_6x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_8x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA

GL_COMPRESSED_SRGB8_ALPHA8_ASTC_8x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_8x8_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_10x5_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_10x6_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_10x8_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_10x10_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_12x10_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_12x12_KHR	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_3x3x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_4x3x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_4x4x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_4x4x4_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_5x4x4_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_5x5x4_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_5x5x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_6x5x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_6x6x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_RGBA_ASTC_6x6x6_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_3x3x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_4x3x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_4x4x3_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_4x4x4_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_5x4x4_OES	GL_UNSIGNED_BYTE	GL_RGBA

GL_COMPRESSED_SRGB8_ALPHA8_ASTC_5x5x4_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_5x5x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_6x5x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_6x6x5_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COMPRESSED_SRGB8_ALPHA8_ASTC_6x6x6_OES	GL_UNSIGNED_BYTE	GL_RGBA
GL_COLOR_INDEX1_EXT	GL_UNSIGNED_BYTE	GL_COLOR_INDEX
GL_COLOR_INDEX2_EXT	GL_UNSIGNED_BYTE	GL_COLOR_INDEX
GL_COLOR_INDEX4_EXT	GL_UNSIGNED_BYTE	GL_COLOR_INDEX
GL_COLOR_INDEX8_EXT	GL_UNSIGNED_BYTE	GL_COLOR_INDEX
GL_COLOR_INDEX12_EXT	GL_UNSIGNED_SHORT	GL_COLOR_INDEX
GL_COLOR_INDEX16_EXT	GL_UNSIGNED_SHORT	GL_COLOR_INDEX
GL_DEPTH_COMPONENT16	GL_UNSIGNED_SHORT	GL_DEPTH_COMPONENT
GL_DEPTH_COMPONENT24	GL_UNSIGNED_INT	GL_DEPTH_COMPONENT
GL_DEPTH_COMPONENT32	GL_UNSIGNED_INT	GL_DEPTH_COMPONENT
GL_DEPTH_COMPONENT32F	GL_FLOAT	GL_DEPTH_COMPONENT
GL_DEPTH_COMPONENT32F_NV	GL_FLOAT	GL_DEPTH_COMPONENT
GL_STENCIL_INDEX1	GL_UNSIGNED_BYTE	GL_STENCIL_INDEX
GL_STENCIL_INDEX4	GL_UNSIGNED_BYTE	GL_STENCIL_INDEX
GL_STENCIL_INDEX8	GL_UNSIGNED_BYTE	GL_STENCIL_INDEX
GL_STENCIL_INDEX16	GL_UNSIGNED_BYTE	GL_STENCIL_INDEX
GL_DEPTH24_STENCIL8	GL_UNSIGNED_INT_24_8	GL_DEPTH_STENCIL
GL_DEPTH32F_STENCIL8	GL_FLOAT_32_UNSIGNED_INT_24_8_REV	GL_DEPTH_STENCIL
GL_DEPTH32F_STENCIL8_NV	GL_FLOAT_32_UNSIGNED_INT_24_8_REV	GL_DEPTH_STENCIL

Acknowledgements

This software uses the FreeImage open source image library. See http://freeimage.sourceforge.net for details.