

FFmpeg Scaler Documentation

Table of Contents

- 1 Description
- 2 Scaler Options
- 3 See Also
- 4 Authors

1 Description# TOC

The FFmpeg rescaler provides a high-level interface to the libswscale library image conversion utilities. In particular it allows one to perform image rescaling and pixel format conversion.

2 Scaler Options# TOC

The video scaler supports the following named options.

Options may be set by specifying *-option value* in the FFmpeg tools. For programmatic use, they can be set explicitly in the `SwsContext` options or through the `libavutil/opt.h` API.

`sws_flags`

Set the scaler flags. This is also used to set the scaling algorithm. Only a single algorithm should be selected.

It accepts the following values:

`'fast_bilinear'`

Select fast bilinear scaling algorithm.

`'bilinear'`

Select bilinear scaling algorithm.

`'bicubic'`

Select bicubic scaling algorithm.

`'experimental'`

Select experimental scaling algorithm.

`'neighbor'`

Select nearest neighbor rescaling algorithm.

`'area'`

Select averaging area rescaling algorithm.

`'bicublin'`

Select bicubic scaling algorithm for the luma component, bilinear for chroma components.

`'gauss'`

Select Gaussian rescaling algorithm.

`'sinc'`

Select sinc rescaling algorithm.

`'lanczos'`

Select Lanczos rescaling algorithm.

`'spline'`

Select natural bicubic spline rescaling algorithm.

`'print_info'`

Enable printing/debug logging.

`'accurate_rnd'`

Enable accurate rounding.

`'full_chroma_int'`

Enable full chroma interpolation.

`'full_chroma_inp'`

Select full chroma input.

`'bitexact'`

Enable bitexact output.

srcw

Set source width.

`srch`

Set source height.

`dstw`

Set destination width.

`dsth`

Set destination height.

`src_format`

Set source pixel format (must be expressed as an integer).

`dst_format`

Set destination pixel format (must be expressed as an integer).

`src_range`

Select source range.

`dst_range`

Select destination range.

`param0, param1`

Set scaling algorithm parameters. The specified values are specific of some scaling algorithms and ignored by others. The specified values are floating point number values.

`sws_dither`

Set the dithering algorithm. Accepts one of the following values. Default value is 'auto'.

'auto'

automatic choice

'none'

no dithering

'bayer'

bayer dither

‘ed’

error diffusion dither

‘a_dither’

arithmetic dither, based using addition

‘x_dither’

arithmetic dither, based using xor (more random/less apparent patterning than a_dither).

alphablend

Set the alpha blending to use when the input has alpha but the output does not. Default value is ‘none’.

‘uniform_color’

Blend onto a uniform background color

‘checkerboard’

Blend onto a checkerboard

‘none’

No blending

3 See Also# TOC

ffmpeg, ffplay, ffprobe, ffserver, libswscale

4 Authors# TOC

The FFmpeg developers.

For details about the authorship, see the Git history of the project ([git://source.ffmpeg.org/ffmpeg](http://source.ffmpeg.org/ffmpeg)), e.g. by typing the command `git log` in the FFmpeg source directory, or browsing the online repository at <http://source.ffmpeg.org>.

Maintainers for the specific components are listed in the file MAINTAINERS in the source code tree.

This document was generated using *makeinfo*.

FFmpeg Scaler Documentation

Table of Contents

- 1 Description
- 2 Scaler Options
- 3 See Also
- 4 Authors

1 Description# TOC

The FFmpeg rescaler provides a high-level interface to the libswscale library image conversion utilities. In particular it allows one to perform image rescaling and pixel format conversion.

2 Scaler Options# TOC

The video scaler supports the following named options.

Options may be set by specifying *-option value* in the FFmpeg tools. For programmatic use, they can be set explicitly in the `SwsContext` options or through the `libavutil/opt.h` API.

`sws_flags`

Set the scaler flags. This is also used to set the scaling algorithm. Only a single algorithm should be selected.

It accepts the following values:

`'fast_bilinear'`

Select fast bilinear scaling algorithm.

`'bilinear'`

Select bilinear scaling algorithm.

`'bicubic'`

Select bicubic scaling algorithm.

`'experimental'`

Select experimental scaling algorithm.

`'neighbor'`

Select nearest neighbor rescaling algorithm.

`'area'`

Select averaging area rescaling algorithm.

`'bicublin'`

Select bicubic scaling algorithm for the luma component, bilinear for chroma components.

`'gauss'`

Select Gaussian rescaling algorithm.

`'sinc'`

Select sinc rescaling algorithm.

`'lanczos'`

Select Lanczos rescaling algorithm.

`'spline'`

Select natural bicubic spline rescaling algorithm.

`'print_info'`

Enable printing/debug logging.

`'accurate_rnd'`

Enable accurate rounding.

`'full_chroma_int'`

Enable full chroma interpolation.

`'full_chroma_inp'`

Select full chroma input.

`'bitexact'`

Enable bitexact output.

srcw

Set source width.

`srch`

Set source height.

`dstw`

Set destination width.

`dsth`

Set destination height.

`src_format`

Set source pixel format (must be expressed as an integer).

`dst_format`

Set destination pixel format (must be expressed as an integer).

`src_range`

Select source range.

`dst_range`

Select destination range.

`param0, param1`

Set scaling algorithm parameters. The specified values are specific of some scaling algorithms and ignored by others. The specified values are floating point number values.

`sws_dither`

Set the dithering algorithm. Accepts one of the following values. Default value is 'auto'.

'auto'

automatic choice

'none'

no dithering

'bayer'

bayer dither

‘ed’

error diffusion dither

‘a_dither’

arithmetic dither, based using addition

‘x_dither’

arithmetic dither, based using xor (more random/less apparent patterning than a_dither).

alphablend

Set the alpha blending to use when the input has alpha but the output does not. Default value is ‘none’.

‘uniform_color’

Blend onto a uniform background color

‘checkerboard’

Blend onto a checkerboard

‘none’

No blending

3 See Also# TOC

ffmpeg, ffplay, ffprobe, ffserver, libswscale

4 Authors# TOC

The FFmpeg developers.

For details about the authorship, see the Git history of the project ([git://source.ffmpeg.org/ffmpeg](http://source.ffmpeg.org/ffmpeg)), e.g. by typing the command `git log` in the FFmpeg source directory, or browsing the online repository at <http://source.ffmpeg.org>.

Maintainers for the specific components are listed in the file MAINTAINERS in the source code tree.

This document was generated using *makeinfo*.