



# darktable 3.8 user manual

# Table of Contents

- 1. Overview 6
  - 1.1. user interface 6
    - 1.1.1. views 6
    - 1.1.2. screen layout 6
    - 1.1.3. filmstrip 8
    - 1.1.4. top panel 8
    - 1.1.5. keyboard shortcuts 8
  - 1.2. supported file formats 9
  - 1.3. sidecar files & non-destructive editing 10
    - 1.3.1. sidecar files 10
    - 1.3.2. importing sidecar files generated by other applications 11
    - 1.3.3. local copies 11
  - 1.4. an introduction to darktable's workflow 12
    - 1.4.1. introduction 12
    - 1.4.2. import & review 12
    - 1.4.3. process 13
    - 1.4.4. export 19
- 2. Lighttable 20
  - 2.1. overview 20
  - 2.2. lighttable view layout 20
  - 2.3. undo/redo 21
  - 2.4. lighttable modes 21
    - 2.4.1. filemanager 21
    - 2.4.2. zoomable lighttable 22
    - 2.4.3. culling 22
    - 2.4.4. full preview 23
  - 2.5. digital asset management 23
    - 2.5.1. collections & film rolls 23
    - 2.5.2. thumbnails 23
    - 2.5.3. star ratings & color labels 24
    - 2.5.4. image grouping 26
    - 2.5.5. metadata and tagging 26
- 3. Darkroom 27
  - 3.1. overview 27
  - 3.2. darkroom view layout 27
  - 3.3. the pixelpipe 28
    - 3.3.1. the anatomy of a processing module 28
    - 3.3.2. the pixelpipe & module order 29
    - 3.3.3. the history stack 31
    - 3.3.4. undo and redo 31
  - 3.4. processing modules 31
    - 3.4.1. module header 31
    - 3.4.2. multiple instances 32
    - 3.4.3. presets 33
    - 3.4.4. module controls 35
    - 3.4.5. curves 37
    - 3.4.6. wavelets 38
    - 3.4.7. deprecated modules 44
  - 3.5. masking & blending 45
    - 3.5.1. overview 45
    - 3.5.2. blend modes 46
    - 3.5.3. masks 49
      - 3.5.3.1. overview 49
      - 3.5.3.2. drawn masks 49
      - 3.5.3.3. parametric masks 52
      - 3.5.3.4. combining drawn & parametric masks 54
      - 3.5.3.5. mask refinement & additional controls 55
      - 3.5.3.6. raster masks 57
  - 3.6. organization 57
    - 3.6.1. overview 57
    - 3.6.2. module groups 58
    - 3.6.3. quick access panel 60
    - 3.6.4. manage module layouts 61
- 4. Tethering 64
  - 4.1. overview 64
  - 4.2. tethering view layout 64
  - 4.3. examples 65
  - 4.4. troubleshooting 65
- 5. Map 67
  - 5.1. overview 67
  - 5.2. map view layout 68
- 6. Slideshow 69
  - 6.1. overview 69
  - 6.2. usage 69
- 7. Print 70
  - 7.1. overview 70
  - 7.2. print view layout 70
- 8. Module Reference 71
  - 8.1. overview 71
  - 8.2. processing modules 72
    - 8.2.1. astrophoto denoise 72
    - 8.2.2. base curve 72
    - 8.2.3. basic adjustments (deprecated) 73
    - 8.2.4. bloom 74
    - 8.2.5. blurs 74
    - 8.2.6. censorize 76
    - 8.2.7. channel mixer (deprecated) 76
    - 8.2.8. chromatic aberrations 78
    - 8.2.9. color balance 78
    - 8.2.10. color balance rgb 81
    - 8.2.11. color calibration 87
    - 8.2.12. color contrast 98
    - 8.2.13. color correction 98
    - 8.2.14. color look up table 98
    - 8.2.15. color mapping 99
    - 8.2.16. color reconstruction 100
    - 8.2.17. color zones 100
    - 8.2.18. colorize 105



8.2.19. contrast brightness saturation	105
8.2.20. contrast equalizer	106
8.2.21. crop	109
8.2.22. crop and rotate (deprecated)	111
8.2.23. defringe (deprecated)	113
8.2.24. demosaic	114
8.2.25. denoise (profiled)	116
8.2.26. diffuse or sharpen	120
8.2.27. dithering	125
8.2.28. exposure	125
8.2.29. fill light (deprecated)	126
8.2.30. filmic rgb	126
8.2.31. framing	136
8.2.32. global tonemap (deprecated)	136
8.2.33. graduated density	137
8.2.34. grain	137
8.2.35. haze removal	138
8.2.36. highlight reconstruction	138
8.2.37. highpass	139
8.2.38. hot pixels	139
8.2.39. input color profile	139
8.2.40. invert (deprecated)	140
8.2.41. lens correction	141
8.2.42. levels	142
8.2.43. liquify	142
8.2.44. local contrast	146
8.2.45. lowlight vision	148
8.2.46. lowpass	148
8.2.47. lut 3D	149
8.2.48. monochrome	149
8.2.49. negadoctor	150
8.2.50. orientation	152
8.2.51. output color profile	153
8.2.52. raw black/white point	153
8.2.53. raw chromatic aberrations	153
8.2.54. raw denoise	154
8.2.55. retouch	154
8.2.56. rgb curve	159
8.2.57. rgb levels	160
8.2.58. rotate and perspective	160
8.2.59. rotate pixels	164
8.2.60. scale pixels	164
8.2.61. shadows and highlights	164
8.2.62. sharpen	165
8.2.63. soften	165
8.2.64. split-toning	165
8.2.65. spot removal (deprecated)	166
8.2.66. surface blur	167
8.2.67. tone curve	167
8.2.68. tone equalizer	168
8.2.69. tone mapping (deprecated)	172
8.2.70. unbreak input profile	172
8.2.71. velvia	173
8.2.72. vibrance (deprecated)	173
8.2.73. vignetting	173
8.2.74. watermark	174
8.2.75. white balance	175
8.2.76. zone system (deprecated)	177
8.3. utility modules	178
8.3.1. darkroom	178
8.3.1.1. clipping warning	178
8.3.1.2. color assessment	179
8.3.1.3. duplicate manager	180
8.3.1.4. gamut check	181
8.3.1.5. global color picker	181
8.3.1.6. guides & overlays	182
8.3.1.7. history stack	182
8.3.1.8. image information line	183
8.3.1.9. mask manager	183
8.3.1.10. module order	186
8.3.1.11. navigation	187
8.3.1.12. raw overexposed warning	187
8.3.1.13. snapshots	187
8.3.1.14. soft proof	188
8.3.2. lighttable	188
8.3.2.1. history stack	188
8.3.2.2. import	190
8.3.2.3. lua scripts installer	193
8.3.2.4. select	193
8.3.2.5. selected images	194
8.3.2.6. styles	195
8.3.2.7. timeline	196
8.3.3. map	196
8.3.3.1. find location	196
8.3.3.2. locations	196
8.3.3.3. map settings	197
8.3.4. print	198
8.3.4.1. print settings	198
8.3.5. shared	199
8.3.5.1. collections	199
8.3.5.2. export	203
8.3.5.3. filmstrip	206
8.3.5.4. focus peaking	207
8.3.5.5. geotagging	208
8.3.5.6. image information	210
8.3.5.7. metadata editor	210
8.3.5.8. recently used collections	211
8.3.5.9. scopes	211
8.3.5.10. tagging	215
8.3.6. tethering	219
8.3.6.1. camera settings	219
8.3.6.2. live view	220
8.3.6.3. session	220
9. Preferences & Settings	221
9.1. overview	221
9.2. general	221
9.3. import	222
9.4. lighttable	223
9.5. darkroom	224
9.6. processing	225

- 9.7. security 227
- 9.8. storage 228
- 9.9. miscellaneous 229
- 9.10. shortcuts 230
- 9.11. presets 236
- 9.12. lua options 237
- 10. Scripting with Lua 238
  - 10.1. overview 238
  - 10.2. basic principles: luarc files 238
  - 10.3. a simple lua example 238
  - 10.4. printing labeled images 238
  - 10.5. adding a simple shortcut 240
  - 10.6. exporting images with lua 241
  - 10.7. building user interface elements 242
  - 10.8. sharing scripts 243
  - 10.9. calling lua from dbus 244
  - 10.10. using darktable from a lua script 244
  - 10.11. lua API 245
- 11. Guides & Tutorials 246
  - 11.1. developing monochrome images 246
  - 11.2. other resources 248
- 12. Special Topics 249
  - 12.1. color management 249
    - 12.1.1. overview 249
    - 12.1.2. display profile 249
    - 12.1.3. rendering method 249
    - 12.1.4. rendering intent 249
    - 12.1.5. darktable's color spaces 250
    - 12.1.6. unbounded colors 252
    - 12.1.7. possible color artifacts 252
    - 12.1.8. darktable's color dimensions 252
  - 12.2. memory 258
  - 12.3. opengl 261
    - 12.3.1. the background 261
    - 12.3.2. how opengl works 262
    - 12.3.3. activating opengl in darktable 262
    - 12.3.4. setting up opengl 263
    - 12.3.5. possible problems & solutions 264
    - 12.3.6. amd/ati devices 265
    - 12.3.7. performance optimization 265
    - 12.3.8. scheduling profile 267
    - 12.3.9. multiple devices 268
    - 12.3.10. opengl still does not run for me 269
  - 12.4. using darktable-chart 269
    - 12.4.1. overview 269
    - 12.4.2. usage 269
    - 12.4.3. source image 270
    - 12.4.4. reference values 271
    - 12.4.5. process 272
    - 12.4.6. making input images for darktable-chart 273
  - 12.5. program invocation 273
    - 12.5.1. darktable 273
    - 12.5.2. darktable-cli 275
    - 12.5.3. darktable-generate-cache 276
    - 12.5.4. darktable-chart 277
    - 12.5.5. darktable-cltest 278
    - 12.5.6. darktable-cmstest 278
    - 12.5.7. purge\_non\_existing\_images.sh 278
  - 12.6. variables 279
  - 12.7. default module order 280
  - 12.8. darktable's color pipeline 287
  - 12.9. midi device support 289
  - 12.10. contributing to dtdocs 291
  - 12.11. translating dtdocs 296

darktable is an open source photography workflow application and raw developer — a virtual lighttable and darkroom for photographers. It manages your digital negatives in a database, lets you view them through a zoomable lighttable and enables you to develop and enhance your raw images.

The source repository for this documentation may be found at <https://github.com/darktable-org/dtdocs.git> . Any feedback relating to this documentation can be provided by creating a [ticket](#) or a pull request against this repository.

This documentation is released under the [GPL 3.0](#) license.

# 1. Overview

## 1.1. user interface

### 1.1.1. views

The functionality in darktable is separated into six different views:

#### **lighttable**

Manage images and collections.

#### **darkroom**

Develop a single image.

#### **map**

Show geo-tagged images on a map and manually geo-tag new images.

#### **print**

Send images to a printer.

#### **slideshow**

Display images as a slideshow, processing them on-the-fly.

#### **tethering**

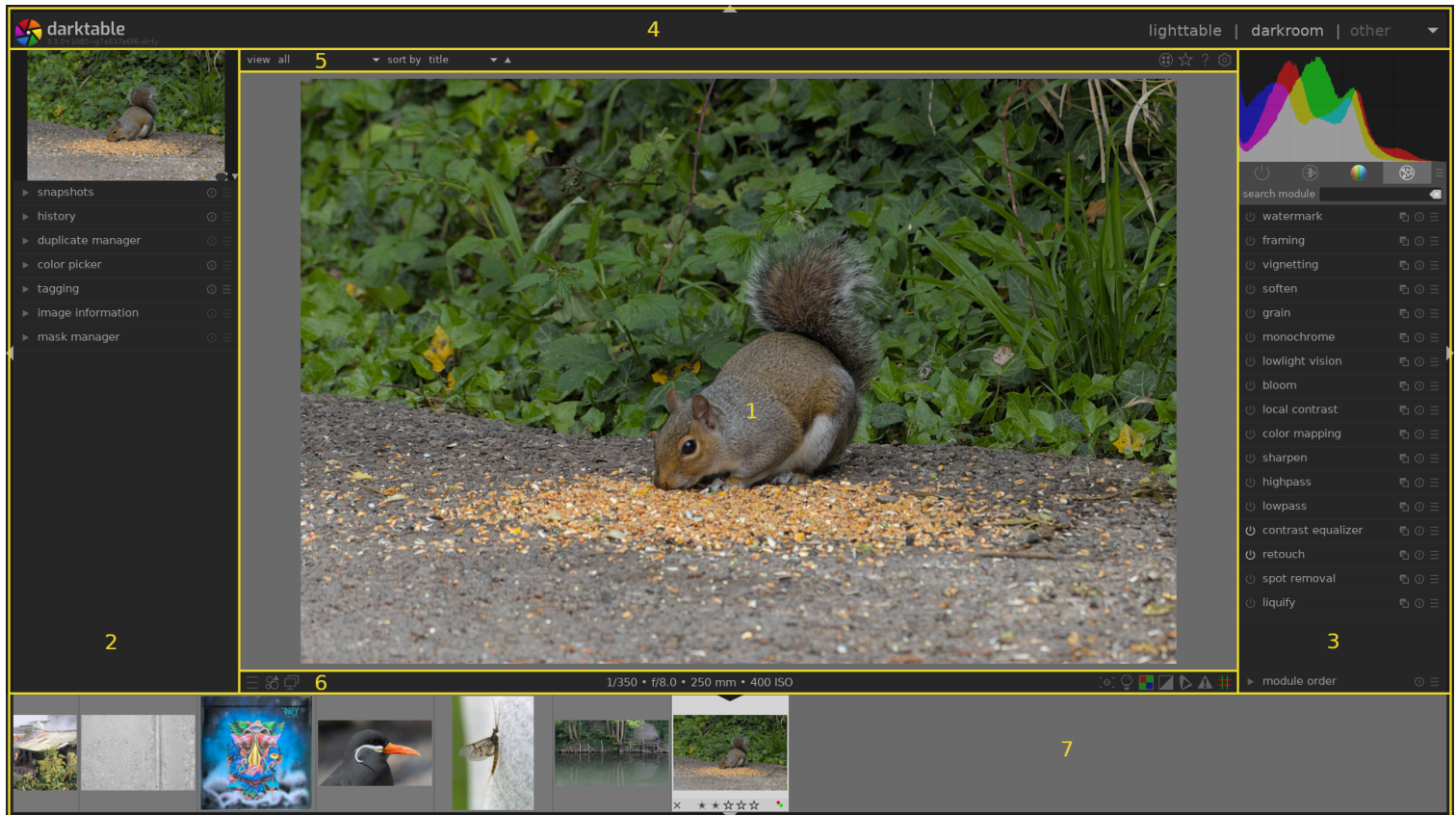
Remotely capture and save images taken with a connected camera.

You can switch between views by clicking the view name at the top of the right-hand panel (the currently active view is highlighted) or by using one of the following keyboard shortcuts:

L	switch to lighttable
D	switch to darkroom
M	switch to map
P	switch to print
S	switch to slideshow
T	switch to tethering

### 1.1.2. screen layout

The layout of all darktable views is similar and consists of a center area with panels at the edges:



1. center area : Contains information and functionality specific to the current view.
2. left panel : Contains modules primarily used to provide information.
3. right panel : Contains modules primarily used for image processing.
4. top banner : Contains information about the current darktable version and allows you to switch between views. Also used by some modules to show additional hints and messages.
5. [top panel](#) : Provides access to global settings and shortcuts
6. bottom panel : Provides access to view-specific settings and shortcuts.
7. [filmstrip / timeline](#) panel : An optional panel that can be enabled at the bottom of the screen to display a timeline (in the lighttable view) or a filmstrip (in other views) of images in the current collection.

## panel size and visibility

The left, right and filmstrip/timeline panels can be resized by dragging their inner borders.

Each of the panels can be expanded or collapsed by pressing the triangle located at the outside edge of the panels. Panel visibility can also be adjusted using keyboard shortcuts, as follows:

TAB	Temporarily expand the centre view to fill the whole window. Press again to return to the previous view.
F11	Toggle fullscreen mode
Shift+Ctrl+t	Toggle the top panel (between the image and the top banner)
Shift+Ctrl+b	Toggle the bottom panel (between the image and the filmstrip/timeline, if shown)
Shift+Ctrl+l	Toggle the left panel
Shift+Ctrl+r	Toggle the right panel
Ctrl+f	Toggle the filmstrip/timeline
Ctrl+h	Toggle the top banner
b	Toggle all borders and panel-collapse controls

**Note:** The size and visibility of the panels are stored independently for each view.

### 1.1.3. filmstrip

The filmstrip, when enabled, is shown at the bottom of the screen (except in the lighttable view, where it is replaced with the [timeline](#) module) and displays the images from the current [collection](#) (set in the lighttable view). You can navigate the filmstrip by scrolling with the mouse wheel.

The filmstrip allows you to interact with images while you are not in the lighttable view. For example, while developing an image in darkroom mode, you can switch to another image to by clicking its thumbnail in the filmstrip. You can also rate and color-classify the images as you do in lighttable, as well as copy & paste the history stack using keyboard shortcuts.

See the [filmstrip](#) module documentation for more information.

### 1.1.4. top panel

The top panel is common to all darktable views and provides access to a number of common functions.



## On the left-hand-side

### view

Choose which images to view in the lighttable/filmstrip, based on star rating and reject status.

### sort by

Choose a property by which to sort the images.

### ▲ sort order

Switch the sort order (ascending / descending).

## On the right-hand-side

### ⊞ grouping

Expand or collapse [grouped images](#) .

### ★ thumbnail overlays

Define what information is overlaid on to thumbnails in the lighttable/filmstrip.

You can define different settings depending on the thumbnail size. See [preferences > lighttable](#) for details of how size delimiters are set.

### ? context-sensitive help

Click this icon and then click on a control element to be directed to the appropriate online help page.

### ⌨ shortcut mapping

Click this icon to enter the [visual shortcut mapping](#) mode to create keyboard/mouse shortcuts.

### ⚙ preferences

View and amend darktable's [preferences & settings](#) .

### 1.1.5. keyboard shortcuts

Much of the functionality in darktable can be controlled via keyboard shortcuts, which can be customised in [preferences > shortcuts](#) .

Press the H key (for help) in any darktable view to show a list of all shortcuts that are applicable to the current view.

## 1.2. supported file formats

darktable supports a huge number of file formats from various camera manufacturers. In addition darktable can read a wide range of low- and high-dynamic-range images - mainly for data exchange between darktable and other software.

In order for darktable to consider a file for import, it must have one of the following extensions (case independent): 3FR, ARI, ARW, BAY, BMQ, CAP, CINE, CR2, CR3, CRW, CS1, DC2, DCR, DNG, GPR, ERF, FFF, EXR, IA, IIQ, JPEG, JPG, K25, KC2, KDC, MDC, MEF, MOS, MRW, NEF, NRW, ORF, PEF, PFM, PNG, PXN, QTK, RAF, RAW, RDC, RW1, RW2, SR2, SRF, SRW, STI, TIF, TIFF, X3F

If darktable was compiled with JPEG2000 support, the following extensions are also recognized: J2C, J2K, JP2, JPC.

If darktable was compiled with GraphicsMagick support, the following extensions are also recognized: BMP, DCM, GIF, JNG, JPC, JP2, MIFF, MNG, PBM, PGM, PNM, PPM.

## camera raw files

darktable reads raw files using the open source library [RawSpeed](#), originally developed by Klaus Post and now maintained as part of the darktable project. The number of supported cameras and file formats is constantly increasing. Most modern camera models are supported, and new ones tend to get added very quickly. It is beyond the scope of this manual to give an exhaustive list.

With the exception of Fujifilm X-Trans cameras, darktable does not decode images from cameras with non-Bayer sensors (e.g. Sigma cameras with the Foveon X3 sensor).

## other image files

darktable natively reads "ordinary" images in JPEG, 8-bit/16-bit PNG and 8-bit/16-bit TIFF format, as well as 16-bit/32-bit floating point TIFF formats.

darktable also reads high dynamic range images in OpenEXR, RGBE and PFM formats.

## 1.3. sidecar files & non-destructive editing

### 1.3.1. sidecar files

darktable is a non-destructive image editor and opens all images in read-only mode. Any data created within darktable (metadata, tags, and image processing steps) is stored in separate *.XMP sidecar* files. These files are stored alongside the original Raw files and allow darktable to store information about the images as well as the full editing history without touching the original raw files. When you import an image into darktable for the first time, an XMP file is automatically generated. The generation of XMP files can be disabled in [preferences > storage > xmp](#) but this is not recommended in normal use.

For a given source image, multiple editing versions, called *duplicates*, can co-exist, sharing the same input image data but each having their own metadata, tags and processing steps. Each duplicate of a given image (named `<basename>.<extension>`) is represented by a separate XMP sidecar file (with a filename constructed in the form `<basename>_nn.<extension>.xmp`, where `nn` represents the version number of that edit). Information for the initial edit – the “duplicate” with version number zero – is stored in the sidecar file named `<basename>.<extension>.xmp`. The version number of each duplicate is displayed in the [image information](#) module in each of darktable’s views.

Your work is automatically synchronised to the sidecar files without the need to press a “save” button. When backing up your data, make sure that you also retain copies of the XMP files, as these are required to fully reconstruct your work in case of a disaster.

In addition to the sidecar files, darktable keeps all image-related data in its library database for fast access. An image can only be viewed and edited from within darktable if its data has first been loaded into the library database. This happens automatically when you first [import](#) an image. If an image is subsequently re-imported, the database will be updated from the contents of its XMP file.

Once an image has been imported into darktable, the database entries take precedence over the XMP file. Subsequent changes to the XMP file by any other software are not visible to darktable – such changes will be overwritten the next time darktable synchronizes the file. On request, darktable can be configured to search for updated XMP files at startup, offering a choice to update the database or overwrite the XMP file where changes are identified. This configuration can be changed in [preferences > storage > xmp](#) .



## 1.3.2. importing sidecar files generated by other applications

When importing an image, darktable automatically checks if it is accompanied by a sidecar file. As well as looking for files named `<basename>.<extension>.xmp` and `<basename>_nn.<extension>.xmp` (darktable's XMP file naming formats) darktable also checks for the presence of a file in the form `<basename>.xmp` (the naming format for Lighroom's XMP sidecar files). Files with the latter naming format will be read by darktable but will not be written to. Once the image has been imported, darktable will generate an additional XMP file using its own naming convention.

At present, darktable is able to load the following metadata from Lightroom-generated sidecar files during the import process:

- tags (including hierarchical tags)
- color labels
- ratings
- GPS information

In addition, darktable has been designed to help migrate some image operations from other specific applications. The aim here is not to make darktable a drop-in replacement for any other software, but rather to help you to recover part of the work you have already invested into your image. It is important to understand that the import process will never give identical results to other software. The underlying processing engines are very different from application to application, and depend a lot on the individual image. In some cases, the results may be similar but often, further adjustment will be required in darktable.

This migration happens automatically when entering the darkroom view, provided that a corresponding XMP sidecar is found.

At present, darktable is able to handle the following development steps from Lightroom-generated XMP files (with the corresponding darktable module in parentheses):

- crop and rotate ( [crop and rotate](#) )
- black level ( [exposure](#) )
- exposure ( [exposure](#) )
- vignette ( [vignetting](#) )
- clarity ( [local contrast](#) )
- tone curve ( [tone curve](#) )
- HSL ( [color zones](#) )
- split-toning ( [split-toning](#) )
- grain ( [grain](#) )
- spot removal ( [spot removal](#) )

## 1.3.3. local copies

Many users have huge image collections stored on extra hard drives in their desktop computer, or on an external storage medium (RAID NAS, external hard drives etc.).

It is a common requirement to develop a number of images while travelling using a laptop and then later synchronize them back to the original storage medium. However, copying images manually from the main storage to the laptop and back is cumbersome and prone to errors. The “local copies” feature of darktable has been designed to directly support these use cases.

You can create local copies of selected images from within the lighttable. Local copies are always used when present, giving continued access to images even if the external storage is no longer connected. At a later point, when your primary storage medium has been reconnected, you can synchronize the XMP sidecar files back to this storage, deleting any local copies. These operations can be found in the [selected images](#) module in the lighttable.

For safety reasons, if local copies exist and the external storage is available, the local XMP sidecars are automatically synchronized at start up.

Local copies are stored within the `$HOME/.cache/darktable` directory and named `img-<SIGNATURE>.<EXT>` (where SIGNATURE is a hash signature (MD5) of the full pathname, and EXT is the original filename extension).

Local copies can be identified in the lighttable view by a white marker on the top right of the thumbnail. In addition, all local copies carry the `darktable|local-copy` tag to allow them to be easily selected.

# 1.4. an introduction to darktable's workflow

## 1.4.1. introduction

This section is intended to provide you with a flavor of what darktable can do and offer a suggested end-to-end processing workflow. New users should start by working through this introduction and following the links provided for further details.

Please note that this is not an exhaustive guide and many useful topics are not covered. You are advised to read through the reference sections of this manual once you are familiar with the basics.

Further information on most of the subjects in this guide are also covered in the [other resources](#) section.

## 1.4.2. import & review

Before you can do anything in darktable you must first add some image files to the library using the [import](#) module in the [lighttable view](#). This will create entries for your images in darktable's library database so that it can keep track of the changes you make. There are three ways to import images, each accessible through buttons in the import module:

### add to library

This option adds images to the library without copying or moving – your original files will stay in their current location and will not be altered. On import, darktable will read the metadata from the image files and any accompanying [XMP sidecar file](#). If an image has already been added to the database, any updates you have made to the sidecar file will be loaded.

### copy & import

Copies the images to the storage location (following the file naming pattern defined in [preferences > import](#)), then adds the *copied* images to the library – the original images are not changed or moved.

### copy & import from camera

Connect the camera to your system with a USB cable (if your camera is auto-mounted by your system, you will need to un-mount it before it can be accessed by darktable). If you don't see your camera listed in the import module, press the "scan for devices" button. Once your camera is detected the import module should offer the ability to *copy & import* images from the camera. Clicking the "copy & import" button physically copies the selected images from the camera into a specified directory (following the file naming pattern defined in [preferences > import](#)) and then adds the copied images to the library.

Once images have been imported, their thumbnails are displayed in the lighttable view, within which you can organize and catalog your imported images – please refer to the [digital asset management](#) section for more information.

The primary use for the lighttable view is to review your images and decide which you would like to edit further and which to discard. The following is a possible culling process to choose which images to edit/delete:

1. Set the lighttable view to only show images with a rating of exactly 1 star using the *view* setting on the [top panel](#). Since darktable assigns a 1 star rating to newly-imported images by default, this will show all of the images you have just imported. You can use the [collections](#) module to further refine your selection if needed.
2. Perform a quick first-level screening of your images: If any images are badly out-of-focus or otherwise unwanted, *reject* them or give them a 0-star rating (by pressing the "R" or "0" keys, respectively, while hovering over the image with your mouse). If you would like an image to pass to the next review phase, press 2 to give it a 2 star rating. Any images that no longer have a 1 star rating will automatically disappear from view. Continue in this manner until no more images remain visible in the lighttable view.
3. Alter the lighttable view to only show images with 2 stars. Go through these images more carefully, and decide whether to promote them to a 3 star rating, or put them back down to a lower rating. Again, proceed until no more images remain in the lighttable view.
4. Alter the lighttable view to only show images with 3 stars. Perform quick edits and experiments on those images in the [darkroom view](#) to decide if they are worthy of further effort. If you are happy with the results of those edits, promote that image to a 4 star rating for final editing.
5. Go through your 4 star images, perform any final edits on them and then [export](#) to see the final result. Increase the rating a final time to 5 stars to indicate that processing is complete.

*If space is at a premium you might consider permanently deleting your rejected or 0-star images. Select these images in the lighttable view and use the 'trash/delete' option in the [selected images](#) module. You should only do this for images you are certain you will never need again.*

## 1.4.3. process

This section is intended to get you comfortable processing images in the darkroom view using a *scene-referred* workflow. You are advised to follow the guidelines provided below, up to the end of the [image processing in 3 modules](#) section and then choose other areas to learn as-and-when you need to use those techniques in your images.

### getting started

#### take a well-exposed photograph

Good image processing techniques start in the camera – a well-exposed image (without blown highlights or heavily crushed blacks) will always make post-processing much more straightforward. Under- or over-exposure can be “fixed” by darktable to some extent but no software can recover information that is not present in the Raw image. Where possible, you are advised to use [exposure to-the-right \(ETTR\)](#) techniques to maximize the amount of data available for processing. As a general rule of thumb, it is safe to under-expose all images by 0.5 to 1 EV (by reducing the ISO sensitivity if possible) even if the in-camera preview looks darker than expected (the preview is not the raw data).

#### scene-referred workflow: a new approach

If you have used other Raw software in the past (or darktable prior to version 3.0) you may notice some significant differences from what you are used to – darktable now uses a *scene-referred* approach for most of its processing modules. This approach is used extensively in cinematography and is known to be much more robust than the traditional *display-referred* approach.

In *display-referred processing* the data from your Raw file is initially compressed into a range that represents pure black as 0 and pure white as 1, fixing mid-gray at 0.5. A tone curve is automatically (and irreversibly) applied to this data to make the image look “good” on your display and subsequent edits are carried out on top of this already highly-modified image data. The cost of *display-referred* is an early loss of the relationship between pixel luminosity and saturation (usually also involving hue shifts), which is responsible for the infamous “HDR look” when the dynamic range increases.

In the real world, “pure black” does not really exist (there is always *some* light) and there is no limit to how bright things can be (so no “pure white” either). *Scene-referred processing* attempts to retain the physical properties of the scene for as long as possible by placing the Raw data on an unbounded linear scale and only compressing the data to the dynamic range of your display after image processing is complete.

In a scene-referred workflow many common tools (tone curves and levels, for example) are no longer useful ways to manipulate the image, since they rely on now-invalid definitions of black, white and gray. Experienced users may need to learn new techniques and discard old ones, but will be rewarded with much more robust and predictable outputs.

---

Set [preferences > processing > auto-apply pixel workflow defaults](#) to “scene-referred” now.

---

Enter preferences by clicking on the gear icon in the [top panel](#) .

#### white balance and color calibration

Most processing software uses a traditional temperature/tint model for adjusting the white balance of an image. In darktable, the [color calibration module](#) provides a much more robust and flexible approach, allowing you to explicitly define the color of the light source. This is particularly useful for scenes illuminated by artificial lighting.

Please note that the [white balance](#) module is still enabled in this approach, but its settings normally should not be altered.

---

Set [preferences > processing > auto-apply chromatic adaptation defaults](#) to “modern” now.

---

## edit in a controlled environment

Image processing should be performed in a controlled environment, lit by a white light source against a background approximating mid-gray, and on a monitor that has been properly calibrated.

While this may not be practical in many home editing environments you *can* control the background colors on your monitor's display. You should set the darktable color scheme to use one of the "grey" themes and use the [color assessment mode](#) when altering tones and colors in your image. Dark themes might look good but, unless you are processing images to be viewed on a cinema screen in a darkened room, they should not be used for photo processing.

---

Set [preferences > general > theme](#) to "darktable-elegant-grey" or "darktable-icons-grey" now.

---

## enter the darkroom

Choose an image to edit from the lighttable view and double-click to load that image into the darkroom view. For now try to choose an image that is well exposed – we will discuss some techniques to recover badly-exposed images later.

In the darkroom view, you will see a list of [processing modules](#) to the right of your image. Each module performs its own processing on the image, in the order shown in the module list, starting at the bottom of the list and moving up to the top. You can think of this like a stack of building blocks where each block builds on the processing performed by the modules below it.

On the left hand side is the [history stack](#) (you may need to expand the module), which shows the order in which adjustments were made to the controls of the various modules. This allows you to undo changes by reverting to an earlier step in the history stack. You will see that a number of modules are applied automatically – these are needed in order to generate a legible image from the Raw data.

It is important to understand the distinction between the order of the modules on the right-hand-side of the screen (which represents the order in which modules are *executed*) and the order of the modules in the history stack (which represents the order in which modules were *modified*).

To the top right is the [scopes](#) module, which shows the spread of tones/colors in your image.

---

If you have previously viewed or edited the image in the darkroom view, start by *discarding history* (click the *reset* button in the [history stack](#) ). This will reapply defaults using your new settings and provide a clean starting point for editing.

---

## why doesn't the raw image look like the JPEG?

*...because you haven't processed it yet*

One of the first things people notice when switching from lighttable to darkroom view is that the image looks different – often flatter and less saturated than that shown in the lighttable view. This is because the darkroom view displays the (mostly unprocessed) Raw image, but the lighttable view initially displays the (in-camera) JPEG preview. Now that you have opened the image in the darkroom view, the lighttable view will update to show the edited version.

Most Raw software goes to great lengths to reproduce the look of standard camera JPEGs out of the box. While this can be useful (if you only want to make very minor adjustments to the camera's rendition of an image) we assume that you are using a Raw editor to make the image your own, and that the camera does not know how to do this. Certainly, if you are using the ETTR techniques mentioned above, the camera JPEG will rarely be close to how you want the final image to look.

The default settings in darktable are therefore intended to provide you with a neutral *starting point* for further editing and nothing more. *We do not intend to change this.*

## module groups

Below the scopes module, at the top right of the screen, is a set of tabs into which darktable's modules are [grouped](#) . If you cannot find a module in one of the tabs you can use the search feature to locate it.

---

---

For the purposes of this guide, click on the hamburger icon (to the right of the tabs) and select the “workflow: scene referred” preset now.

---

## image processing in 3 modules

The following basic adjustments are fundamental to scene-referred editing and will be required, to some extent, on the majority of images. You can usually produce a good-looking image with these steps alone.

As you will be adjusting the tones and colors of the image, start by enabling [color assessment mode](#) (press Ctrl+B) and perform the following edits on the zoomed-out image while in this mode.

1. **Set overall image brightness:** First, set the overall (average) brightness of the image (the mid-gray point) by adjusting the *exposure* slider in the [exposure](#) module. This is a purely artistic setting and should be defined based on your intent – for example, for a high-key image you will set the average brightness to be lighter than for a low-key image. The color assessment mode provides you with two reference points to assist with this by surrounding the image with a white frame against a middle-gray background.

At this point, don't worry if the brightest parts of your image lose detail – this can be recovered in the next step.

---

**Note:** The [lens correction](#) module can also affect the image brightness so you may want to consider enabling it before adjusting exposure.

---

2. **Set white and black points:** The next two steps use the [filmic rgb](#) module to define how the tones in your image will be mapped to the dynamic range of your display. Start by setting the *white* and *black* relative exposure sliders in the [scene tab](#). These are purely technical settings, defining white and black relative to the mid-gray point you set in the previous step. If your image contains tones you want to treat as pure white or pure black you can use the color pickers beside the sliders to set these values (using the maximum and minimum brightness of the image). Otherwise set the values manually using the color assessment frames as a reference.
3. **Adjust the contrast:** Now move to the [look tab](#) in *filmic rgb* (for now we will skip the [reconstruct tab](#)). Enable the [look only view](#) at the top of the module to see a representation of the filmic tone curve, which consists of a straight section in the middle (used to set the contrast of the mid-tones) and curved sections at the top and bottom (where the shadows and highlights are compressed to fit the dynamic range of the display).

The *contrast* slider changes the slope of the straight section (the mid-tone image contrast), the *latitude* slider changes its length and the *shadows/highlights balance* slider changes its position. There is a lot of give-and-take involved here – if you want to increase the contrast of the mid-tones, you must sacrifice contrast in the shadows/highlights and vice versa. The default settings of this module are tuned to work for the majority of images but you should experiment with these sliders to understand how they affect the image.

---

**Note:** The highlight compression in the *filmic rgb* module can cause detail to be lost in the highlights. You can mitigate this to some extent by reducing the white relative exposure, adjusting the shadows/highlights balance or changing the contrast in highlights setting in the [options tab](#). The [tone equalizer](#) module can also be used to reduce the relative brightness of the sky.

---

4. **Color preservation:** The tone mapping in the filmic module attempts to redistribute the tones in your image without affecting color reproduction. While the default color preservation algorithm works for most images, you are encouraged to experiment by changing the *preserve chrominance* setting in the [options tab](#) if you do not like how the colors appear.
5. **Saturation:** Your image will probably not look very colorful at this point. You can adjust the global saturation of the image using the [color balance rgb](#) module. The “basic colorfulness” [preset](#) should provide you with generally-reasonable defaults, but you are encouraged to experiment further with these settings as required.

---

**Note:** This guide assumes that the white balance of the image has been correctly captured by your camera. If this is not the case, you may need to make some corrections in the [color calibration](#) module first.

---

You can now switch off *color assessment mode* by pressing Ctrl+B again.

## other processing techniques

With practice the above workflow can quickly provide you with a reasonable-looking image, though most will need some additional work before they are ready for export. The following sections are intended to provide a brief outline of some more image processing techniques in darktable, with links to the relevant reference sections for more information.

As a general rule, you should begin with the basic steps outlined in the previous section, then perform [corrections](#) and finish with [creative adjustments](#).

## corrections

### color calibration

Traditional white balance correction attempts to ensure that whites and grays are really neutral ( $R = G = B$ ) and doesn't really try to manage the impact on other colors. The [CAT tab of the color calibration module](#) extends this treatment to handle the remainder of the color range and works in a color space designed specifically for chromatic (color) adaptation. As with traditional white balance controls you can select a patch of neutral gray in your image to calculate the white balance, or use a selection of other automatic and manual methods. The default settings use the white balance from the image's Exif data and are usually sufficient.

If you need to make adjustments in the *color calibration* module, you may want to also revisit any saturation corrections you made earlier in the *color balance rgb* module.

### correct lens distortions

All lenses introduce some artifacts (distortion, chromatic aberrations, vignetting) to the image. The [lens correction](#) module can correct many of these issues for a wide variety of lenses. The [chromatic aberrations](#) and [raw chromatic aberrations](#) modules can also be used to handle chromatic aberrations for lenses that are not (or only partially) supported by *lens correction*. In most cases simply enabling the *lens correction* module will auto-detect your lens and auto-apply all available corrections.

If you decide to use the *lens correction* module, it should be enabled at the start of your edit, before you adjust exposure, since vignetting correction can alter the overall brightness of your image.

### reduce noise / retain detail

At a pixel level there is a trade-off to be made between the retention of fine details and the reduction/removal of sensor noise. In most situations, a small amount of noise is perfectly acceptable and will not be noticeable except when you zoom in to 100%. At this scale you are not viewing the image at a realistic size – even when represented on a large monitor or print, noise that is obvious at high zoom factors will be virtually invisible on the final image. However, some modules further along the image pipeline (especially those that increase local contrast) may end up exaggerating any noise that is present so, again, there are trade-offs to be made.

The first module you can use to manage this is [demosaic](#), which controls how the single-color (R, G or B) pixels in your Raw file are converted to image pixels combining all three colors. Each demosaic algorithm has its own trade-offs between retaining fine detail and reducing noise. The default demosaic algorithm (RCD) usually provides a reasonable compromise.

Demosaic algorithms can only do so much to manage noise in your image. The [denoise \(profiled\)](#) module is individually tuned for a number of common camera sensors and can be used to reduce or remove pixel noise. As with demosaic you should alter the settings until you are happy with the balance of denoising vs fine-detail reproduction. The default settings are usually sufficient.



## sharpness and local contrast

A number of modules can be used to adjust the local contrast and sharpness of your image. Most of these modules aim to enhance the apparent contrast of edges and do not add “real” sharpness (they are not the same as lens deconvolution). You should take care when using these modules as most of them can introduce artifacts (such as halos) when settings are pushed too far:

- the [contrast equalizer](#) module allows you to adjust contrast, limiting the effect to certain feature sizes. For example you can use it to increase the contrast of fine details without impacting larger-scale objects, or vice versa
- the [diffuse or sharpen](#) module offers a number of presets for sharpening, lens deblurring and the addition of local contrast
- the [local contrast](#) module provides a simpler interface for quickly adding local contrast to your images – just enabling the module or selecting one of the presets is often all that is required
- the [sharpen](#) module is intended to re-introduce sharpness that was removed by your camera’s anti-alias filter (if present) and can be enabled by default in [preferences > processing](#) . The methods listed above are usually preferred to this legacy module.

As with the modules mentioned in the previous section, you should take care when adding contrast to small-scale objects – an image viewed at 100% is not a realistic representation of your final edit and local contrast adjustments are usually better judged when zoomed out.

## reconstruct blown highlights

While a well-exposed image will make post-processing much easier, darktable provides some tools handle blown highlights, in some cases even reconstructing lost colors or structure.

The [highlight reconstruction](#) module provides a very basic attempt to manage blown highlights. By default, if any color channel is clipped, this module will clip the other channels, making the pixel white – this prevents darktable from showing unnatural colors in the highlights. However it can be altered to attempt to reconstruct the lightness or color using any unclipped channels. While this can re-introduce some detail, beware of the unrealistic colors that can result.

A better alternative is to disable *highlight reconstruction* and instead use the [reconstruct tab on the filmic rgb module](#) . As with the *highlight reconstruction* module, filmic is able to reconstruct detail based on unclipped channels. Where this is not possible, it can also blend blown highlights smoothly with the rest of the image or pull colors from adjacent pixels. When incorporating this step into your workflow, you should do it at the same time as other *filmic rgb* edits, performing adjustments in each tab of the module in turn, from left to right.

## adjust angle and perspective

The [rotate and perspective](#) module can be used to adjust the angle of the image or to simulate the functionality of a tilt/shift lens by altering the perspective, making converging horizontal and/or vertical lines parallel (keystone correction). This latter technique is most commonly used for architectural photography. If you just want to correct the angle of the horizon you can do this by right-clicking and dragging along the horizon line.

## remove spots and unwanted objects

Use the [retouch](#) module to remove unwanted objects by replacing pixels with detail from elsewhere in the image. This module also offers powerful techniques for removing large-scale objects (such as spots or blemishes) while leaving fine-scale details (like hairs and follicles) intact. The most common use for this module is to remove dust spots from images or blemishes from skin.

## remove atmospheric haze

There are two methods for removing atmospheric haze in darktable. The [haze removal](#) module provides a much simpler interface, but the “dehaze” preset in the [diffuse or sharpen](#) module can provide more flexibility when needed.

## creative adjustments

### crop and frame

Use the [crop](#) module to crop your image and the [framing](#) module to surround your image with a colored frame. Both modules can be set to use a predefined or custom aspect ratio – for example, you could place a square-cropped image into a 3:2 frame.

### dodge and burn

Dodging and burning is a traditional darkroom technique to add and remove brightness from an image. There are two recommended ways to achieve this

- If you want to selectively dodge or burn only certain objects you can apply a [new instance](#) of the [exposure](#) module using a [drawn mask](#) to isolate the effect to the required area of the image (see also the [mask refinement](#) section for more information). Move the exposure slider to alter the brightness of the masked area.
- If you wish to dodge or burn areas with a similar brightness (for example, to brighten the shadows or darken the highlights) you should use the [tone equalizer](#) module.

### convert to monochrome

darktable provides a number of ways to remove the color from your image. The most flexible method is to use the [gray tab of the color calibration module](#). A number of film emulation presets are available in this module to provide you with a starting point.

See the [developing monochrome images](#) section for details of other techniques.

### color grading

The [color balance rgb](#) module is your one-stop-shop for controlling the colors in your image. Adjustments can be isolated to the shadows, highlights and mid-tones, or applied at a global level.

## other important topics

### re-use common module settings

If you find yourself using the same module parameters repeatedly, you can create [presets](#) containing your favorite settings. If you use the same settings on every image, you can also make presets apply automatically to new images. For example, you may find yourself adding the same exposure settings to every image taken by a certain camera – in this case you can create a preset that automatically applies those corrections only to images from that camera.

You may also have groups of module settings that you commonly apply only to certain types of image. You can use [styles](#) to apply multiple module settings at once to a selection of images.

### perform adjustments locally

Most darktable modules can either be applied to the whole image, or restricted to parts of the image using [drawn and parametric masks](#).

### control darktable with other input methods

You don't need to use the darktable UI to make adjustments to your images. Most of the functionality in darktable can also be controlled using shortcuts defined with your keyboard/mouse and even with other input devices such as game controllers and MIDI devices. See the [shortcuts](#) section for details.



## 1.4.4. export

darktable is a non-destructive editor, which means that all changes are recorded in the library database (with a backup stored in an XMP sidecar file), and the original Raw file is left untouched. You therefore need to [export](#) images in order to bake your edits into an output file that can be distributed outside of darktable.

1. Choose an export scenario.

The export module offers many options, but by far the most common use is to “save a developed raw image as a JPEG”. You can either export the currently-edited image directly from the darkroom view or select one or more images from the lighttable view and export them all at once.

2. Select which images to export (if you are in the lighttable view), open the export module, set *target storage* to “file on disk” and select a location to save your images - by default, they will be exported to a “darktable\_exported” directory within the directory that contains your Raw file(s). Choose a “file format” of JPEG and keep the default settings.

3. Click the “export” button to save your processed images in the selected location.

**Note:** *While JPEG is useful for most purposes, if you wish to perform further edits in a raster editor like GIMP or Krita, it is normally better to export in tiff format.*

## 2. Lighttable

### 2.1. overview

The lighttable view allows you to view and manage your image collection.

The centre view contains thumbnails of your images – how they are displayed depends on which [mode](#) you are working in.

While the mouse is over an image thumbnail or images are selected, there are a number of actions you can perform with keyboard shortcuts:

0 – 5	set the rating of the image if an image has 1 star and you hit the 1 key, the image will become unrated
R	rate the image as "rejected"
F1 – F5	set a color label
Ctrl+C	copy the history stack
Ctrl+V	paste the copied history stack
D	open in darkroom view for developing
W	fully zoom into the current image while the key is pressed
Ctrl+W	fully zoom into the current image and show areas in focus

### 2.2. lighttable view layout

#### left panel

From top to bottom:

##### [import](#)

Import images from the filesystem or a connected camera.

##### [collections](#)

Filter the images displayed in the lighttable center panel – also used to control the images displayed in the [filmstrip](#) and [timeline](#) modules.

##### [recently used collections](#)

View recently used collections of images.

##### [image information](#)

Display image information.

##### [lua scripts installer \(optional\)](#)

Install lua scripts.

#### right panel

From top to bottom:

##### [select](#)

Select images in the lighttable using simple criteria.

##### [selected images\(s\)](#)

Perform actions on selected images.

##### [history stack](#)

Manipulate the history stack of selected images.

##### [styles](#)

Store an image's history stack as a named style and apply it to other images.

##### [metadata editor](#)

Edit metadata for selected images.

**tagging**

Tag selected images.

**geotagging**

Import and apply GPX track data to selected images.

**export**

Export selected images to local files or external services.

## bottom panel



From left to right:

**star ratings**

Apply star ratings to images.

**color labels**

Apply color categories to images.

**mode selector**

Choose a lighttable mode.

**zoom**

Adjust the size of thumbnails.

**enable focus-peaking mode**

Highlight the parts of the image that are in focus.

**set display profile**

Set the display profile of your monitor(s).

## 2.3. undo/redo

Most changes made within the lighttable are recorded and can be reverted to a previous state. This includes modifications to color labels, ratings, geo-localization, tags, metadata, orientation, copy/paste of history, image duplication, or application of a style. Note that the facility to undo/redo actions is unlimited in the number of steps while in the lighttable view, but it is reset each time you switch to a different view.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

## 2.4. lighttable modes

### 2.4.1. filemanager

In the default mode images are displayed in a grid with an adjustable number of images per row.

### controls

**zoom**

The number of images in each row can be altered using the slider in the bottom panel, or by holding Ctrl while scrolling over the center view with your mouse.

**navigate**

You can navigate through the images using the arrow keys (←/→/↑/↓) or by scrolling with your mouse. Press the Home key to scroll to the top of the collection, the End key to scroll to the bottom, and PageUp/PageDown to scroll up/down by a page.

---

**select**

You can select the image under the pointer by clicking on its thumbnail or by pressing Enter. A range of images can be selected by clicking on the first image and then Shift+clicking on the last one. Images can be added or removed from a selection by Ctrl+clicking on their thumbnails or by pressing Spacebar.

## 2.4.2. zoomable lighttable

The *zoomable lighttable* mode provides an alternative way to navigate large collections of images, but with some similarities to the filemanager mode (described below).

### controls

**zoom**

Scroll with the mouse wheel to zoom in and out of the lighttable (compared to Ctrl+scroll in [filemanager](#) mode).

Zooming the thumbnails does not change the number of thumbnails per row, so the lighttable can exceed the visible area on all sides.

**navigate**

Hold down the left mouse button and drag to move the lighttable around and navigate through your collection.

**select**

As with the filemanager mode, you can select the image under the pointer by clicking on its thumbnail or by pressing Enter. A range of images can be selected by clicking on the first image and then Shift+clicking on the last one. Images can be added to or removed from a selection by Ctrl+clicking on their thumbnails or by pressing Spacebar.

---

**Hint:** you may find that image thumbnails are slow to load when zooming quickly through a large collection. One way to speed up the navigation is to generate a cache containing all the thumbnails using the [darktable-generate-cache](#) command.

---

## 2.4.3. culling

Culling mode allows you to display images side by side for easy comparison.

### controls

**zoom**

In culling mode, you can zoom into images (up to 100%) by holding Ctrl while scrolling with the mouse wheel.

Pan within zoomed images with click+drag.

By default, zooming and panning are synchronized between all visible images. If you want to zoom or pan only a specific image, add the Shift modifier to the above actions.

**navigate**

Use the mouse wheel or arrow keys (←/→) to scroll through your collection.

### modes

There are two different culling modes, which define how many images are shown at the same time: “fixed” and “dynamic”. Switch between these while in culling mode by pressing the “<” key.

**fixed mode**

The number of images displayed is always the same, independent of the selection length. This number can be set with a slider on the bottom panel.

In this mode, you will navigate through all selected images. If no selection is set (or if only one image is selected), you will navigate through all images.

The default keyboard shortcut to enter culling in fixed mode is X.

---

### dynamic mode

All of the selected images are shown. If no selection is set (or if only one image is selected) the last value from fixed mode is used.

The default keyboard shortcut to enter culling in dynamic mode is Ctrl+X.

---

**Hint:** To enhance performance when loading zoomed images, you can enable ( [preferences > processing > cpu/gpu/memory > enable disk backend for full preview cache](#) ). Bear in mind that this can occupy a lot of disk space.

---

## 2.4.4. full preview

From any of the lighttable modes, you can display a fully-zoomed preview of the image that is currently under the mouse pointer by pressing and holding down W. This is useful to more closely inspect an image while rating and selecting images.

Pressing and holding Ctrl+W fully zooms into the image and also identifies any regions of sharpness in the image that may indicate image focus. For this tool to work the input image needs to hold an embedded JPEG thumbnail, which is the case for most raw files.

Regions in the image with a high level of sharpness are indicated with red borders. If no such regions are found, any regions of moderate sharpness are identified with a blue border. Note that this is not the same as the [focus peaking](#) indicator, which is another way to identifying areas of sharpness within an image.

Sometimes pressing W or Ctrl+W may not appear to have any effect - in such cases, click on the image thumbnail and press the corresponding key again.

If you want the full preview to stay in place without having to hold the W key, you can enable sticky preview mode by pressing F. In sticky preview mode, you can zoom and pan within the image in a similar way to the [culling](#) mode. Press F or ESC to return to the original view.

## 2.5. digital asset management

### 2.5.1. collections & film rolls

A collection is a set of images matching a given selection criteria.

The most basic kind of collection is a film roll, which contains all of the images that have been imported from a specific folder on disk. Whenever you [import](#) images from the filesystem, those images are organized in a film roll whose name is derived from that of their parent folder.

You can easily construct other kinds of collection based on various image attributes (Exif data, filename, tags etc.) in the [collections](#) module. Multiple criteria can be logically combined to narrow or extend your collection.

darktable retains a list of the most recently used collections for quick access. These can be accessed from the [recent collections](#) module.

### 2.5.2. thumbnails

Each image in the current collection is represented by a thumbnail in the lighttable view and filmstrip module. A cache of the most recently used thumbnails is stored in a file on disk and loaded into memory at startup. The size of this cache can be adjusted in [preferences > processing > cpu/gpu/memory](#) .

## thumbnail creation

A thumbnail is created when an image is imported into darktable for the first time, after an image has been modified in the darkroom, or when revisiting an image whose thumbnail is no longer available.

When an image is imported for the first time darktable can either try to extract an embedded thumbnail from the input image (most raw files contain these, usually in JPEG format) or process the raw image itself using default settings. You can define how darktable obtains its thumbnails in [preferences > lighttable > thumbnails](#) .

Extracting an embedded thumbnail from the input image is usually very fast. However, these thumbnails have been generated by the raw converter of the camera and do not represent darktable's "view" of that image. You will notice the difference as soon as you open the image in the darkroom mode, at which point darktable replaces the thumbnail with its own internally processed version.

After import darktable automatically generates thumbnails for new images as they are needed. When importing a large set of new images, thumbnail generation can slow down navigation in the lighttable view. Alternatively you may terminate darktable and generate the thumbnail cache separately by running [darktable-generate-cache](#) . This program will generate all missing thumbnails in one go.


As the thumbnail cache has a pre-defined maximum size it will eventually get filled up. If new thumbnails are subsequently added, old thumbnails are dropped from the cache. However, darktable will keep all thumbnails on disk if the corresponding disk backend option is activated in [preferences > processing > cpu/gpu/memory](#) . Access to the thumbnails in this secondary cache is slower than the primary cache, but still much faster than reprocessing thumbnails from scratch. The size of the secondary cache is limited only by the available disk space.

Thumbnails are never removed from the secondary cache. You can manually clean the secondary cache by recursively deleting all images in the `$HOME/.cache/darktable/mipmaps-xyz.d` folder (where xyz denotes an alphanumeric identifier of the cache). After clearing the secondary cache you can simply allow darktable to re-generate thumbnails as needed, or you can generate all thumbnails in one go with [darktable-generate-cache](#) .

If you choose not to activate the disk backend and select too small a cache size, darktable may become unresponsive, you may experience continuous regeneration of thumbnails when you navigate your collection or flickering of thumbnail images. A good choice of cache size is 512MB or higher (see [memory](#) for more information).

All thumbnails are fully color managed. Colors are rendered accurately on screen as long as your system is properly set up to hand over the right monitor profile to darktable. For more information see the [color management](#) section.

## skulls

If for some reason darktable is unable to generate a thumbnail, it displays an image of a skull  instead. Don't panic!

There are three main reasons that this could happen:

- *Missing image file*: darktable remembers all images it has ever imported, as long as they have not been removed from your database. If darktable wants to create a thumbnail but is not able to open the input file, a skull is displayed instead. Users are advised to remove images from the database using the [selected images](#) module before physically removing them from disk. Alternatively you may occasionally run the script [purge\\_non\\_existing\\_images.sh](#) from darktable's toolset to clean-up your database.
- *Invalid image format*: While the extension of an image may seem to be supported by darktable, its contents could be either an unsupported image format or a corrupt file.
- *Low memory*: If darktable runs out of memory while generating a thumbnail, it will warn you and display a skull. This can happen if darktable is run with sub-optimal settings, especially on a 32-bit system. See [memory](#) for more information.

### 2.5.3. star ratings & color labels

Star ratings and color labels help you to sort and rank images according to your own criteria. An image's star rating and color labels can be displayed over thumbnails in the lighttable view and filmstrip module.

## star ratings

You can give an image a rating from zero to five stars. Whenever you import images, each image receives a default rating which you can define in the [import](#) module. You can also mark an image as “rejected”.

There are several ways to change a rating. While hovering the cursor over an image thumbnail, you can press a number key 0 - 5 to define the number of stars, or press R to “reject” an image. This is probably the fastest way to rate your images on first inspection of a film roll.

You can also directly click on the star icons that are overlaid on the thumbnails or in the bottom panel. Click the x to reject.

As rejecting an image removes the currently-applied star rating, you can undo the rejection by clicking x or pressing R again.

Similarly you can click the first star for a second time to reset the image rating to unranked, or zero stars. This behavior can be changed in [preferences > lighttable](#) .

To rate multiple images at once, select those images in the lighttable or filmstrip and then press the appropriate shortcut key, or click the desired star rating in the bottom panel of the lighttable view.

You can filter images by star rating in the [top panel](#) .

## color labels

Color labels are another way to classify images, and can be used as an alternative to star ratings or to work alongside them. Each image can carry any combination of one or more color labels (red, yellow, green, blue, or purple).

You can set the color labels for a single image by hovering your cursor over the thumbnail and pressing the function keys F1 - F5, which correspond with the labels in the order given above.


To toggle the color labels of one or more images, select the desired images in the lighttable or filmstrip and then press the appropriate shortcut key or click the corresponding color button in the bottom panel. To remove all labels from the selected images, press the gray button.

You can filter images by color label in the [collections](#) module.

## 2.5.4. image grouping

Grouping images helps to improve the structure and clarity of your image collection when displayed in the lighttable view.

You can combine images into a group by selecting them and clicking the “group” button in the [selected image\(s\)](#) module, or by pressing Ctrl+G. Likewise, you can remove selected images from a group by clicking the “ungroup” button, or pressing Ctrl+Shift+G. Duplicated images are automatically grouped together. Similarly, if you import multiple images from the same directory, having the same base name, but different extensions (eg. IMG\_1234.CR2 and IMG\_1234.JPG), those images automatically form a group.

Images that are members of a group are denoted by a group icon  in their thumbnails.

This icon also appears as a button, in the top panel of the lighttable view, that can be used to toggle grouping on and off. If grouping is off, all images are displayed as individual thumbnails. If grouping is on, the images in a group are represented by a single thumbnail image (the group leader). If you press the group icon in the group leader’s thumbnail, that group is expanded (click a second time to collapse). If you then expand another group, the first group collapses.

An expanded group in the filemanager mode of lighttable view is indicated by an orange frame that appears as soon as your mouse pointer hovers over one of the images. This frame surrounds all images in the group.

You can define which image is considered to be the group leader by clicking on the group icon of the desired image while that group is expanded. The group icon is shown only if grouping mode is enabled, so to change the group leader, you need to first enable grouping, expand the appropriate group and finally click the group icon of the desired “group leader” image. The current group leader is shown in a tooltip when you hover over the group icon of an image.

If you collapse an image group and then enter darkroom mode (e.g. by double-clicking on the thumbnail), the group leader image will be opened for developing.

Image groups are also a convenient way to protect an existing history stack against unintentional changes. If you have just finalized an image and want to protect its current version, simply select the image, click “duplicate” in the selected images panel, and make sure that grouping is switched on and that the group is collapsed. Now, whenever you open the image group again in the darkroom, only the group leader will be altered. The underlying duplicate will remain unchanged.

---

**Note:** “duplicating images” only generates a copy of an image’s history stack, stored in another small XMP file. There is still only one raw file.

---

## 2.5.5. metadata and tagging

darktable allows you to store additional information about your images to allow them to be more easily searched and grouped. This information is stored in darktable’s database and XMP sidecar files and can also be included within exported images.

### metadata

Metadata (e.g. title, description) is free-format text that usually differs for each image. You can add metadata to images in the [metadata editor](#) module.

### tagging

Tags are usually shared between multiple images and are used to categorise and group them. You can add tags to images in the [tagging](#) module.



## 3. Darkroom

### 3.1. overview

The darkroom view is where you develop your images. The center panel contains the image currently being edited.

#### zoom

Middle-click on the center panel cycle between “fit to screen”, 1:1 and 2:1 zoom.

Alternatively you can zoom between 1:1 and “fit to screen” by scrolling with your mouse. Scroll while holding the Ctrl key to extend the zoom range to between 2:1 and 1:10.

### 3.2. darkroom view layout

#### left panel

From top to bottom:

##### [navigation](#)

Navigate and zoom the center view.

##### [snapshots](#)

Take and view snapshots for comparison with the current edit.

##### [duplicate manager](#)

View and manage duplicates.

##### [global color picker](#)

Select and display color information taken from parts of the image.

##### [tagging](#)

Manage tags.

##### [image information](#)

Display information about the current image.

##### [mask manager](#)

View and edit drawn shapes.

##### [export](#)

Export selected images to local files or external services.

#### right panel

From top to bottom:

##### [scopes](#)

A graphical depiction of the image’s light levels and colors. This module can be moved to the left-hand panel if desired (see [preferences > miscellaneous > position of the scopes module](#) ).

##### [module groups](#)

Select module groups (if enabled).

##### [search module](#)

Search for a module (if enabled).

##### [processing modules](#)

The modules used to process an image.

##### [module order](#)

Choose the order in which processing modules are executed in the pixelpipe.

## bottom panel



From left to right:

### [presets](#)

Quick access menu for module presets. You can manage the contents of this menu by selecting “manage quick presets list...”.

### [styles](#)

Quick access menu for styles.

### [second darkroom window](#)

For multi-monitor setup, allows the user to display a second image on another screen.

### [focus peaking](#)

Toggle focus peaking mode.

### [color assessment](#)

Toggle the ISO12646 color assessment view.

### [raw overexposed warning](#)

Toggle raw overexposed indicators (right-click for options).

### [clipping warning](#)

Toggle clipping warnings (right-click for options).

### [soft proof](#)

Toggle softproofing overlays (right-click for options).

### [gamut check](#)

Toggle gamut checking (right-click for options).

### [guides & overlays](#)

Left-click to switch global guide overlays on/off and right-click to change the guide settings, including the color of all on-image drawing (masks, crop guides etc.)

You can also enable the [filmstrip](#) module at the bottom of the screen to allow you select and interact with the currently selected collection in the [lighttable](#) view.

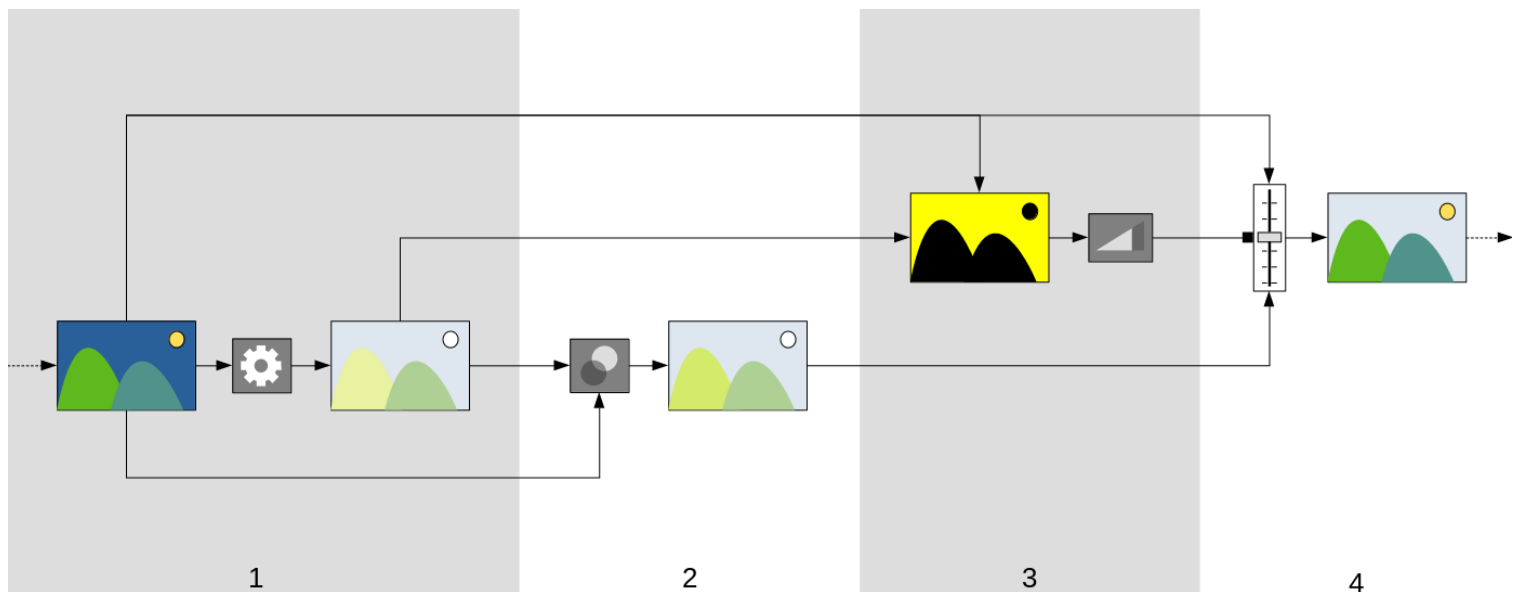
## 3.3. the pixelpipe

### 3.3.1. the anatomy of a processing module

The basic element of image processing in darktable is the [processing module](#) . In order to process a raw image a number of such modules act on the input image in sequence, each performing a different *operation* on the image data. For those familiar with Adobe Photoshop, the concept of a *processing module* in darktable is analogous to that of an *adjustment layer* in that both make an incremental adjustment to the image, building on top of the adjustments that came before.

[Utility modules](#) are also provided by darktable, however these are not directly involved in image processing, instead providing a GUI that allows you to manage your images, tag them, export them etc.

Every processing module acts independently of the others, but all modules perform their processing in a similar manner:



1. Receive the *module input* from the last executed module and perform an *operation* on it to produce the *processed output*. This *operation* is different for every [processing module](#) .
2. Combine the *module input* and *processed output* using a [blending operator](#) to produce the *blended output*. If no blending is performed, the output of this step is the same as the *processed output*.
3. Generate a *mask*, which defines an *opacity* for each pixel in the image. The *opacity* is later used to control how strongly the module's operation is applied to each part of the image.

You may define your own mask by drawing shapes over the image or by using pixel properties from the *module input* or *processed output* (see [masks](#) for details). This mask may be further modified with a global opacity setting, which affects every pixel equally.

If no drawn/parametric mask is used, the output of this step is a mask where every pixel has the same opacity (governed by the global opacity setting). If no opacity is defined (no blending is performed) a global opacity of 1.0 (or 100%) is assumed.

4. Combine the *module input* and *blended output* pixel-by-pixel using the *mask* as a mixing operator, to produce the *final output*. Where the mask opacity is 100%, the *final output* is the *blended output* for that pixel. Where the mask opacity is 0 the final output is the *module input* for that pixel. An intermediate opacity combines the *blended output* and *module input* proportionally. The *final output* is passed to the next module for further processing.

Steps 2 and 3 are optional and not supported by all modules. For example, the [demosaic](#) module must be applied to the entire raw file in order to produce a legible image so it does not make sense to mask or blend its output.

Each of the above steps is defined in more detail in subsequent sections.

### 3.3.2. the pixelpipe & module order

The ordered sequence of [processing modules](#) operating on an input file to generate an output image is known as the "pixelpipe".

The order of the pixelpipe is represented graphically by the order in which modules are presented in the user interface - the pixelpipe starts with a RAW image at the bottom of the module list, and applies the processing modules one by one, piling up layer upon layer of processing from the bottom up, until it reaches the top of the list, where it outputs the fully processed image.

---

**Note:** The order in which processing modules are executed exactly matches the order in which the modules appear in darktable's user interface. **Changing the order of the modules in the user interface changes how your image is processed.**

---

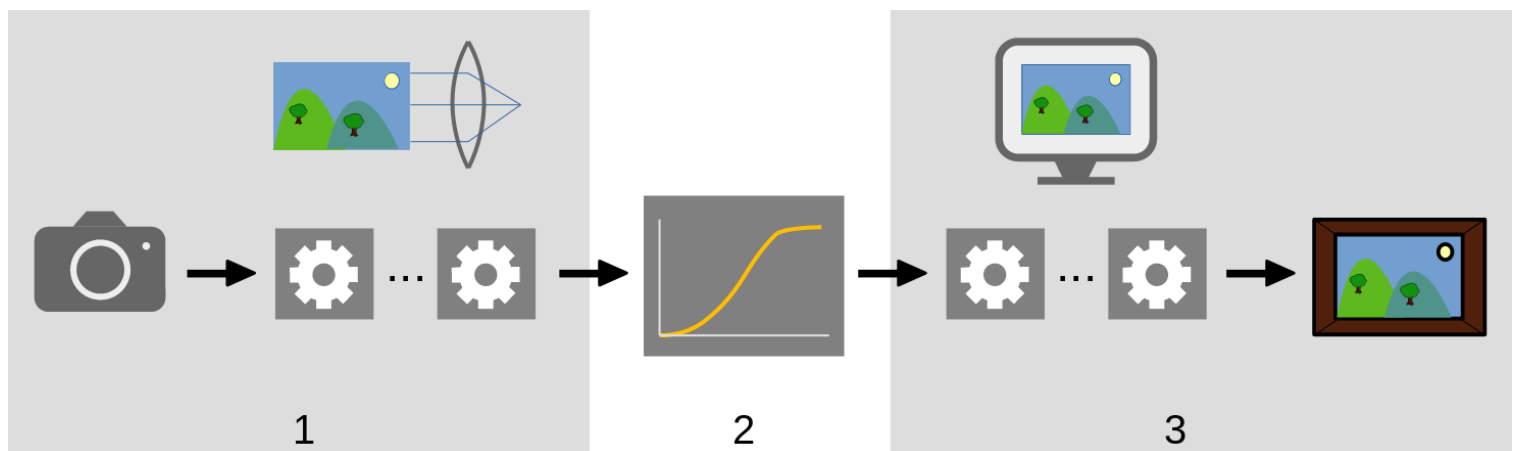
## module order and workflows

The order in which modules are executed within the pixelpipe has been carefully chosen to give the best output quality. In previous versions of darktable it was not possible to change the module order. However, there are a number of very specific use cases where the movement of some modules within the pixelpipe is advised.

One of the main reasons to change the module order came about with darktable version 3.0, which introduced the new *scene-referred* way of working. Version 3.2 formalised this by introducing the *display-referred* and *scene-referred* workflows, which are controlled by the [preferences > processing > auto-apply pixel workflow defaults](#) setting. Starting with version 3.6, *scene-referred* workflow is now the official recommended (and default) way to use darktable.

The *scene-referred* workflow attempts to perform as many operations as possible in a linear RGB color space, only compressing the tones to fit the output medium (with a non-linear tone mapping) at the end of the pixelpipe. This has the advantage of being a more physically-realistic space to do transformations than the traditional *display-referred* workflow, which attempts to perform operations in a non-linear *perceptual* color space. Honoring the physical realism (rather than the perceptual realism) makes it much easier to produce predictable processing algorithms with a minimum of artifacts.

The following diagram should help you to understand the difference between these workflows:



1. *Scene-referred* modules process linear data that is proportional to the amount of light collected by the camera at the scene. The dynamic range of an image in the scene-referred section of the pixelpipe is often larger than that of the display medium.
2. At some point in the pixelpipe, these pixel values are compressed by a non-linear tone mapping operation into a smaller dynamic range more suitable for display on a monitor or a print.
3. The remaining modules operate in the non-linear *display-referred* section of the pixelpipe to produce the final output image.

## display-referred workflow

Prior to version 3.0 darktable's workflow was *display-referred* (auto-apply pixel workflow defaults = "display-referred") and this option is still provided as a legacy mode. In this workflow, the [base curve](#) or [filmic rgb](#) module performs tone mapping early in the pixelpipe and most other darktable modules operate on image data in the compressed *display-referred* space.

Selecting the display-referred workflow enables the legacy (pre-darktable-3.0) module order and automatically switches on the [base curve](#) module for new images.

Pixel data within the *display-referred* space is non-linear and is not a physically realistic representation of the original scene. This can lead to various artifacts with some modules, hence the creation of the (now default) *scene-referred* workflow.

## scene-referred workflow

*Scene-referred* workflow (auto-apply pixel workflow defaults = “scene-referred”) was introduced as part of darktable 3.0. The module order was entirely rearranged to place the [filmic rgb](#) and [base curve](#) tone mapping modules much later in the pixelpipe. This means that most modules now operate in *linear rgb* space with only a few modules remaining within the non-linear *display-referred* space. Within this workflow it is now recommended that the majority of image processing takes place using the modules up to and including [filmic rgb](#). Operations in this section of the pixelpipe, being truly linear, are much more physically realistic and produce fewer artifacts.

Selecting the scene-referred workflow enables the *v3.0* module order and automatically enables the [exposure](#) and [filmic rgb](#) modules with some presets designed to act as a reasonable starting point for scene-referred editing.

## changing module order

It remains highly recommended that users not change the order within the pixelpipe for a number of reasons:

- The sequence of modules has been selected with great care in order to give highest output quality. Changes to the sequence often worsen the result rather than improving it.
- Some processing modules simply don’t make sense if they are shifted in the pixelpipe. For example, [highlight reconstruction](#) needs to be performed on raw data before [demosaic](#), which itself needs to be performed before any [input color profile](#) can be applied. For this reason it is still not possible to move some of the modules that are placed early in the pixelpipe.
- Most processing modules are designed to work within a specific color space (see the [color management](#) section for more details). Full flexibility would require modules to support different parallel algorithms depending on the color space they are working in, which would drastically increase complexity.

Despite the general recommendation to leave the pixelpipe order alone, it is possible to move modules within the pixelpipe by holding Ctrl+Shift and dragging and dropping the desired module to a new location. This should only be done by experienced users who understand the impact this will have on the image.

The module order can be manually changed back to either the *v3.0* or *legacy* versions using the [module order](#) module, which can also be used to define your own custom module order presets.

### 3.3.3. the history stack

The *history stack* stores the entire editing history for a given image, in the order in which those edits were applied. It is saved to darktable’s library database and the image’s XMP sidecar file and persists between editing sessions.

Each time a processing module is enabled, disabled, moved or amended a new entry is added to the top of the *history stack*.

The history stack can be queried and modified within the [history stack](#) module in the darkroom.

---

**Note:** The history stack is not a representation of the order in which the modules are **executed** but the order in which they were **amended**. The execution order is represented by the order of the modules in the right-hand panel.

---

### 3.3.4. undo and redo

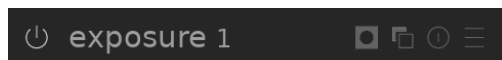
While you are editing your image, darktable records all of the modifications you make to that image. This means that it is possible to undo and redo changes to recover a previous editing state. Note that the undo/redo facility is unlimited in the number of steps while editing an image, but is reset each time the darkroom is switched to a new image.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

## 3.4. processing modules

### 3.4.1. module header

At the top of each processing module is the *module header*.



Click on the module name to expand the module and display the parameters that control its operation.

By default darktable will only allow one processing module to be expanded at a time – if you click the header of another module, the previously-opened module’s controls are collapsed. If you want to expand more than one module, you may expand further modules by Shift+clicking on the header and all previously expanded modules will remain open. This behaviour can be reversed via a setting in [preferences > darkroom](#) .

---

**Note:** Expanding a module does not cause it to be activated. See below for how to activate modules.

---

The module header contains the following controls in order from left to right:

#### on/off button

Click to toggle the module on or off. Some modules are essential for image processing and cannot be disabled (though their parameters may be amended). Similarly, some modules are not applicable for certain types of image and cannot be enabled.

Ctrl+click on the on/off button to toggle whether the module has focus. The focus state is usually used to activate any overlays that a module places over the image to control its functionality. For example, the [crop](#) module only shows the composition and crop guide lines on the image if it has focus. Modules are automatically given focus when expanded.

#### module name

The module name consists of a description of the module’s operation (which cannot be changed) followed by the module’s instance name (which can). By default the first instance of a module has an empty instance name. If you create additional instances, the name of each new instance is initiated with a unique integer. For example, the second created instance of an exposure module will be automatically named exposure 1.

Ctrl+click on a module’s name to manually amend its instance name.

#### mask toggle

This icon will appear in the header whenever a [mask](#) is active on a module. Hover over the icon to see what type of mask is enabled. Click it to display the current mask as a yellow overlay over a black-and-white version of the image. Solid yellow indicates an opacity of 100%; a fully visible gray background image (without yellow overlay) indicates an opacity of 0%. This toggle button can be disabled in [preferences > darkroom > show mask indicator in module headers](#) .

#### multiple instance menu

This menu allows you to create, delete, move and rename module instances. Right-click on this icon to directly create a new instance of the module. See the [multiple instances](#) section for more information.

#### reset

Click to reset all module controls to their default values. Ctrl+click to reapply any automatic [presets](#) for the module – if no automatic presets are applicable for this module, Ctrl+click will simply reset to default values (same as click).

#### presets menu

This menu allows you to apply, create and edit module presets. See the [presets](#) section for more information.

The visibility of the four icons to the right of the module name can be controlled in [preferences > darkroom > show right-side buttons in processing module headers](#) .

## 3.4.2. multiple instances

Many of darktable’s modules can be applied more than once in the pixelpipe. Each instance of a module behaves independently, taking its input from the module below it in the pixelpipe and delivering its output to the next module.

As with the base instance of a module, all instances can be moved independently within the pixelpipe either by holding Ctrl+Shift while dragging & dropping or by choosing “move up” or “move down” in the *multiple instances* drop-down menu.

Instances can be renamed by Ctrl+clicking on the module name.

## typical use cases

There are many occasions where it makes sense to have a module apply more than once in the pixelpipe. Here are some typical use cases.

- The [exposure](#) module can be used in combination with [masks](#) to lighten or darken parts of an image. A separate instance may be created to modify each masked region of the image.
- You may wish to handle luma and chroma noise independently. This can be accomplished by generating two instances of your chosen denoising module and using the first one only on luma (by selecting [blend mode](#) “lightness”) and the second one only on chroma (by selecting blend mode “color”).

---

**Note:** Each instance also adds to the workload of your pixelpipe. Generating too many instances – especially of the more demanding modules – will cause noticeable slow-down.

---

## managing multiple instances

Click on the *multiple instance menu* in the [module header](#) to display a menu with the following options. Right-click on the menu icon to create a new instance directly (same action as clicking on the “new instance” option of the menu).

### **new instance**

Create a new instance of the current module with all of its parameters reset to their default values. The ‘instance name’ is automatically set to a unique integer so that it can be distinguished from its parent.

### **duplicate instance**

Create a new instance of the current module with all of its parameters inherited from the current instance. As with ‘new instance’ the ‘instance name’ is automatically set to a unique integer.

### **move up/down**

Move the instance up or down in the pixelpipe

### **delete**

Remove the current instance. This option is not available if only one instance is present.

### **rename**

Rename the current instance. See the [history stack](#) section for more details on how the instance name impacts copying and pasting history stacks.

## 3.4.3. presets

Presets allow you to store commonly-used module settings for future use. Some modules already come with pre-defined (internal) presets and you may also define your own (user-defined). Both internal and user-defined presets can be shown by clicking the *presets menu* in the [module header](#) .

Most of the functionality described here applies to processing modules only. However, presets can also be used with some utility modules. When used with utility modules, the functionality to auto-apply or auto-show presets based on image Exif data is not available.

Please note that, for processing modules, the saved preset also includes the active state of the module. You can use this to create your own default settings, which you can activate on-demand. Simply set your desired defaults, disable the module, and save the preset.

## the presets menu

The presets menu will contain one or more of the following entries depending on the presets that are defined or selected for the current module:

### **preset list**

A list of the presets available for the current module. The currently selected preset (if any) is shown in **bold** and with a small check mark beside it.

### **edit this preset**

If a preset has been selected, edit the selected preset (see below).

### **delete this preset**

If a preset has been selected, delete the selected preset.

**update preset [name]**

Update the named preset to match the module's current parameters.

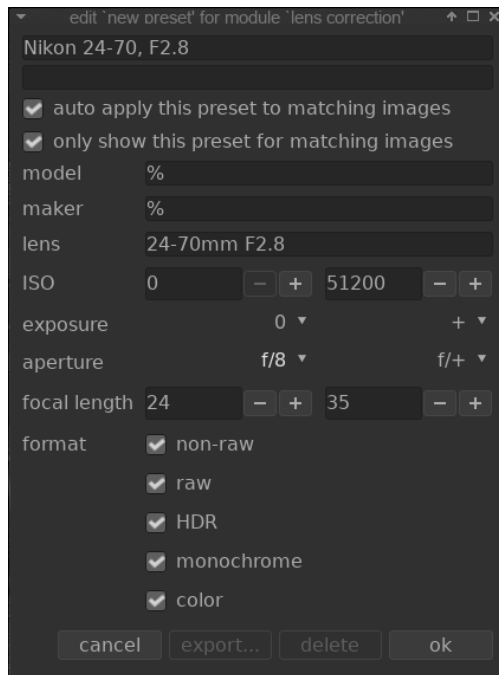
**store new preset**

Create a new preset using the module's current parameters.

Click on a preset name to apply the preset to the current instance of the module. Right-click on a preset name to create a new instance of the module and apply the selected preset to it. You can also apply a preset at any time while you are in the darkroom using a keyboard shortcut – if you have assigned one (see [preferences > shortcuts](#) ).

## creating and editing presets

When creating or editing presets, the following dialog is shown:



## controls

**name**

The name of the preset

**description**

A searchable description for the preset (optional)

**auto apply this preset to matching images (processing modules only)**

Check this box to automatically apply this preset to matching images when they are opened in the darkroom for the first time (you can reapply such automatic presets by Ctrl+clicking on the *reset* button in the [module header](#) ). Additional controls will appear to allow you to define which images the preset will be applied to based on image Exif data (see below).

For example, if you want a preset to be applied to all images from a specific camera leave all fields at default values except for “model”. Leave all fields unchanged to auto-apply a preset to all images.

The example dialog above sets up following rules: if the lens name matches, the aperture is greater than or equal to f/8 and the focal length is between 24 and 35mm the preset will be automatically applied.

The [image information](#) module displays the camera model and lens name for each image. Use this to ensure you have the correct spelling.

**only show this preset for matching images (processing modules only)**

Check this box to automatically show the preset in the preset menu, using the same set of filters.



## filter criteria

The following criteria can be used to auto-apply or auto-show presets for processing modules:

### **model**

A pattern to be matched against the Exif field that describes your camera model; use % as wildcard.

### **maker**

A pattern to be matched against the Exif field that describes the maker of your camera; use % as wildcard.

### **lens**

A pattern to be matched against the Exif field that describes your lens; use % as wildcard.

### **ISO**

Only apply the preset if the ISO value of your image lies within the given range.

### **exposure**

Only apply the preset if the exposure time of your image lies within the given range; set + as the upper value to match arbitrarily long exposures.

### **aperture**

Only apply the preset if the aperture of your image lies within the given range; set f/0 as the lower value to match arbitrarily open apertures; set f/+ as the upper value to match arbitrarily closed apertures.

### **focal length**

Only apply the preset if the focal length of your image lies within the given range (from 0 to 1000).

### **format**

Only apply the preset to certain types of image. Check a box to include files matching a criterion; uncheck the box to exclude those files. Choose from "raw", "non-raw", "HDR", "monochrome" and "color".

---

**Note:** If you deselect both "raw" and "non-raw" the filter will not apply to any files (since all supported files are either raw or non-raw). Similarly if you deselect both "monochrome" and "color".

---

## managing presets

Both user-defined and internal presets can be viewed and managed from within [preferences > presets](#) .

---

**Note:** If you create a user-defined preset with the same name as a built-in preset, your preset will override the built-in version, which will no longer be accessible.

If you delete a preset that has the same name as one of the built-in presets, then your user preset will be deleted, and that preset name will no longer appear in the preset menu at all. The next time you start darktable, the corresponding built-in preset will once again become visible.

---

## 3.4.4. module controls

This section describes how to interact with processing module controls.

### sliders

Sliders offer five different methods of interaction, depending on the level of control you require.

#### **left-click**

Click anywhere in the slider area to set the value. You can also click and drag to change it. You don't have to aim for the triangle or even the line – you can click anywhere in the entire height of the slider, including the label.

#### **mouse wheel**

Hover over the slider with your mouse, then use your mouse wheel to adjust the value.

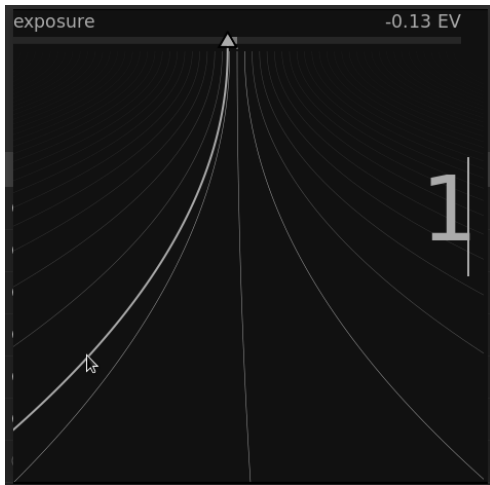
#### **keyboard arrow keys**

When the slider has focus you can hover over the slider with your mouse, then use your keyboard's arrow keys (←/↓ and →/↑) to adjust the value. In order to give focus to the widget without changing the current value you can right-click, then right-click again.

---

### right-click

When your mouse is over a slider right-clicking enables a multi-functional pop-up below the slider for fine control with your mouse or numerical entry using the keyboard.



A bent line extending from the triangular marker moves with your mouse. The closer your mouse pointer is to the triangular marker the coarser the control you have over the value; the further away from the triangular marker the finer your control. Left-click with your mouse to accept the new value and hide the pop-up.

Alternatively you can type in a new value using your keyboard and commit by hitting the Enter key. You may even supply the new value in the form of an arithmetic expression which darktable will calculate for you – the previous value is referenced as “x”.

### double-click

Double-click on a slider or its label to reset to the default value.

In addition, the precision of mouse-wheel and arrow-key adjustments can be altered:

- hold down the Shift key while adjusting to *increase* the step size by a factor of 10.
- hold down the Ctrl key while adjusting to *decrease* the step size by a factor of 10.


Both of these multipliers can be amended in the `$HOME/.config/darktable/erc` file:

```
darkroom/ui/scale_rough_step_multiplier=10.0
darkroom/ui/scale_precise_step_multiplier=0.1
```

## comboboxes

Click on a combobox to show a list of available options which you can click to select. Occasionally the selection list will open close to the bottom or top of the screen meaning that only some of the items are visible – simply scroll with your mouse wheel to bring up the full list. Alternatively, you can also use the mouse wheel and keyboard arrow keys to select an option, or start typing to filter the combobox entries.

## color pickers

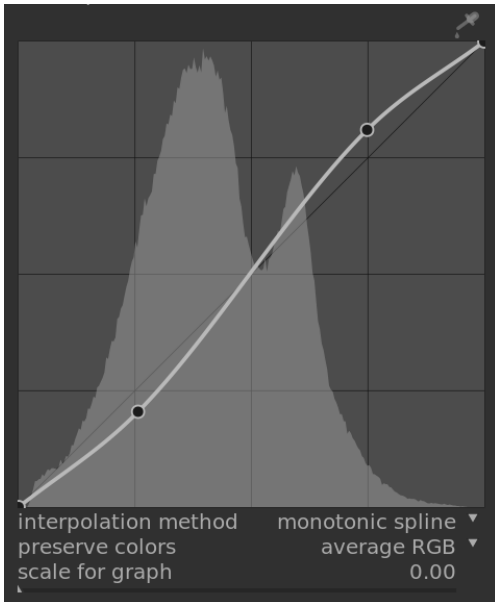
A number of modules allow parameters to be set using color pickers (identified by the  icon). These use a standard interface and most can operate in either point or area mode. Point mode can be activated by clicking on the color picker icon. Area mode is activated with Ctrl+click or right-click.

## keyboard shortcuts

Module parameters can also be adjusted using keyboard shortcuts. See [preferences > shortcuts](#) for more information.

## 3.4.5. curves

The [base curve](#), [tone curve](#) and [rgb curve](#) modules use curves to control the tones in the image. These modules have some common features that warrant separate discussion.



## nodes

In their default state, curves are straight lines, defined by two anchor nodes at the top-right and bottom-left of the graph. You can move the nodes to modify the curve or generate new nodes by clicking on the curve. Ctrl+click to generate a new node at the x-location of the mouse pointer and the corresponding y-location of the current curve – this adds a node without the risk of accidentally modifying the curve. Up to 20 nodes can be defined per curve. To remove a node, click on it and drag it out of the widget area.

## curve controls

The following controls are common to two or more of the above processing modules and are therefore discussed separately here. Please see the individual module documentation for discussion of any additional controls.

### interpolation method

*tone curve and rgb curve only*

Interpolation is the process by which a continuous curve is derived from a few nodes. As this process is never perfect, several methods are offered that can alleviate some of the issues you may encounter.

- *cubic spline* is, arguably, the most visually pleasing. Since it gives smooth curves, the contrast in the image is better enhanced. However, this method is very sensitive to the nodes' position, and can produce cusps and oscillations when the nodes are too close to each other, or when there are too many of them. This method works best when there are only 4 to 5 nodes, evenly spaced.
- *centripetal spline* is designed specifically to avoid cusps and oscillations but, as a drawback, it will follow the nodes more loosely. It is very robust, no matter the number of nodes and their spacing, but will produce a more faded and dull contrast.
- *monotonic spline* is designed specifically to give a monotonic interpolation, meaning that there will be none of the oscillations the cubic spline may produce. This method is most suitable when you are trying to build an analytical function from a node interpolation (for example: exponential, logarithm, power, etc.). Such functions are provided as presets. It is a good trade-off between the two aforementioned methods.

### preserve colors

If a non-linear tone curve is applied to each of the RGB channels individually, then the amount of tone adjustment applied to each color channel may be different, and this can cause hue shifts. The *preserve colors* combobox provides different methods of calculating the “luminance level” of a pixel in order to minimise these shifts. The amount of tone adjustment is calculated based on this luminance value, and then this same adjustment is applied to all three of the RGB channels. Different luminance estimators can affect the contrast in different parts of the image, depending on the characteristics of that image. The user can therefore choose the estimator that provides the best results for the given image. Some of these methods are discussed in detail in the **preserve chrominance** control in the [filmic rgb](#) module. The following options are available:

- *none*
- *luminance*
- *max RGB*
- *average RGB*
- *sum RGB*
- *norm RGB*
- *basic power*

### scale for graph

*tone curve and base curve only*

The scale allows you to distort the graph display so that certain graphical properties emerge to help you draw more useful curves. Note that the scaling option only affects the curve display, not the actual parameters stored by the module.

By default, a “linear” scale is used (defined by a scale factor of 0). This scale uses evenly spaced abscissa and ordinates axes.

A logarithmic scale will compress the high values and dilate the low values, on both the abscissa and the axis of ordinates, so that the nodes in lowlights get more space on the graph and can be controlled more clearly.

Increase the ‘scale for graph’ slider to set the base of the logarithm used to scale the axes. This allows you to control the amount of compression/dilation operated by the scale. If you draw purely exponential or logarithmic functions from identity lines, setting this value defines the base of such functions.

## 3.4.6. wavelets

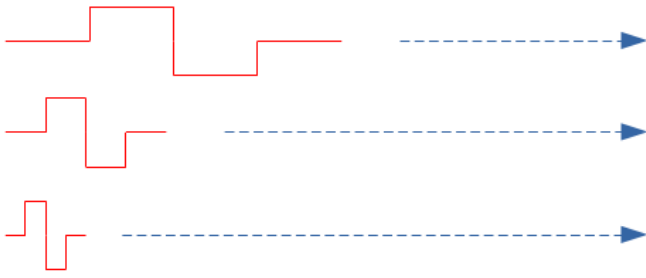
Wavelets are used to separate (or *decompose*) an image into a number of distinct *layers*, each containing a different level of detail. After decomposing an image in this way, a module can limit its processing to one or more of these detail layers, and then piece the layers back together again at the end to form its output. This allows us to be surgical about which features in the image we wish to impact when working with a module.

Some of the operations that darktable can perform in this way are:

- noise removal (in the [denoise \(profiled\)](#) , [raw denoise](#) and [contrast equalizer](#) modules)
- contrast adjustment (in the [contrast equalizer](#) module)
- blurring or removal of unwanted detail (in the [retouch](#) module)

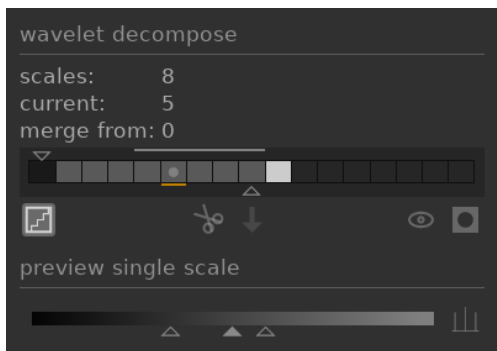
## theory

A wavelet is an oscillating mathematical function that starts and ends at zero. The following diagram shows some simple wavelets of differing sizes.



These wavelet functions are used to scan and decompose an image using a mathematical operation called *convolution*. This picks out details from the image that are on a similar scale to the size of a given wavelet, and builds a number of these detail layers, each corresponding to a different wavelet scale.

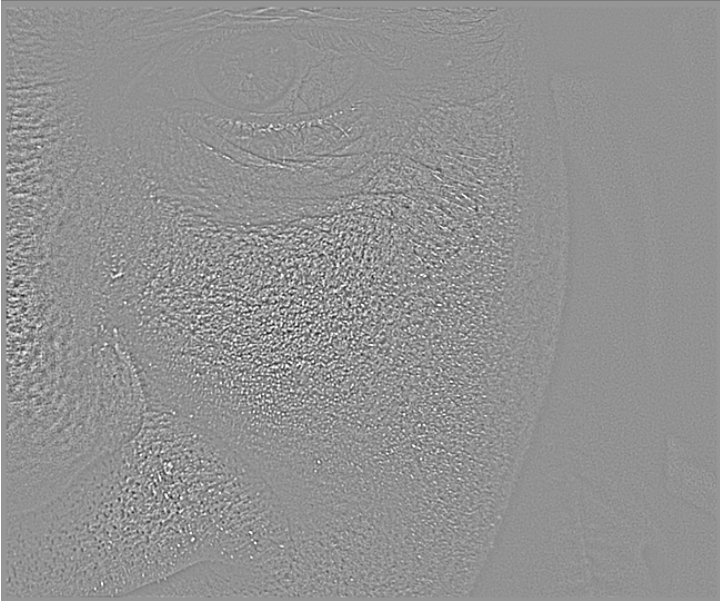
The following is an example where detail layers have been extracted from the above image. In this case, the images were produced using the [retouch](#) module, splitting the image into 8 different layers, and using the module's "display wavelet scale" button to visualise each of the detail layers:



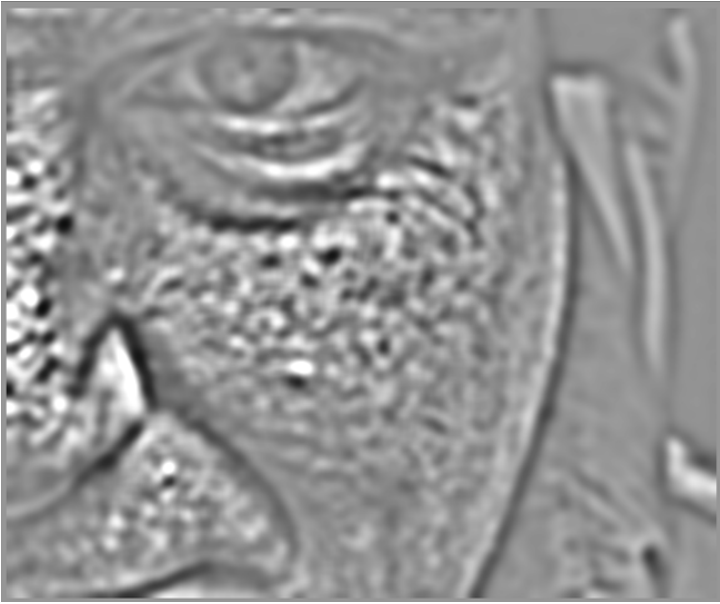
The bars in the *wavelet decompose* section indicate the layers that have been extracted at different wavelet scales. The darkest rectangle at the left represents the entire image (before decomposition) and the gray boxes each represent one of the decomposed layers. Clicking on the staircase icon below the bar graph enables the layer visualisation overlay so that you can see what the currently selected layer looks like.

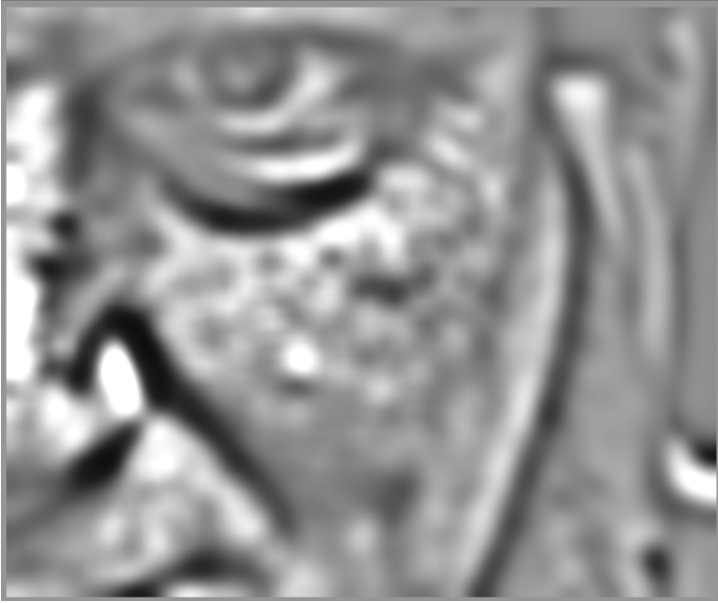
Let's take a look at some of the layers generated for the above image.

At scale #2, the image contains only very fine details, including the model's eyebrows, eye lashes and the pores of his skin. It doesn't include the coarser details of the image, because those details have been extracted to other layers:



At scales #5 and #6 we begin to see larger and larger features:





By scale #8 we only see very high-level features such as the overall shape of the model's nose, eye and cheek:

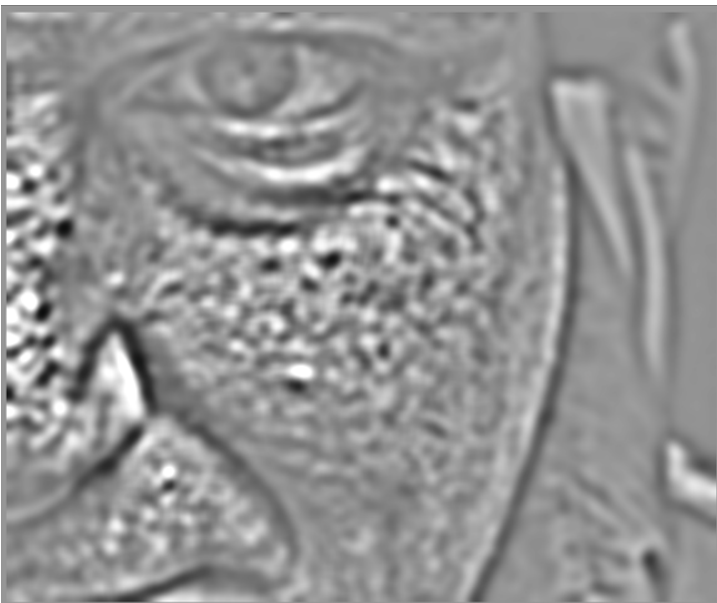
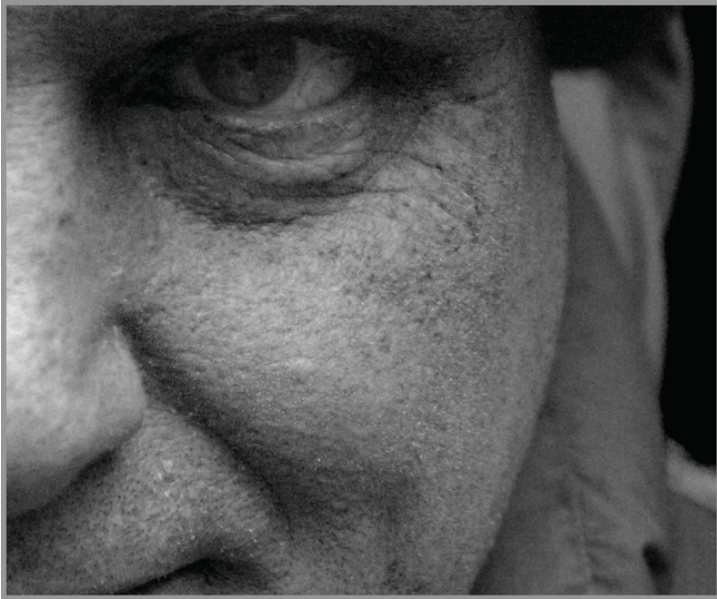


## why use wavelets?

Suppose, in the above example, that we wanted to smooth out some of the blotchiness in the model's skin, without losing any of the underlying small-scale texture. Wavelet decomposition makes this a trivial operation - we can simply use the retouch module to apply a Gaussian blur to only the 'blotchy' detail layer(s), leaving all other layers untouched. Once the adjustment is complete, the retouch module simply recombines the adjusted layer with the remaining untouched layers to produce the final image.

The sequence of images below shows (1) The original image; (2) The layer (scale 5) that we wish to blur; and (3) The final image after the scale 5 layer has been blurred and the layers recombined:









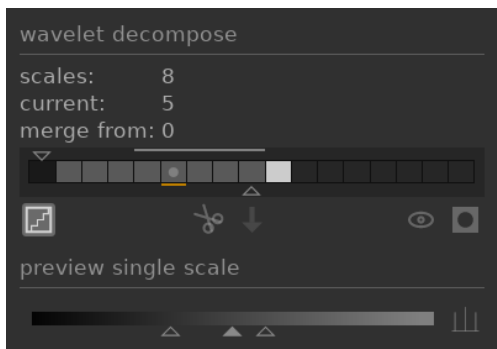
As you can see, the large-scale skin blotches have been removed, but the small-scale details remain untouched.

## interacting with wavelet scales

There are two methods by which processing modules allow you to modify their operation using wavelet scales.

### wavelet decomposition

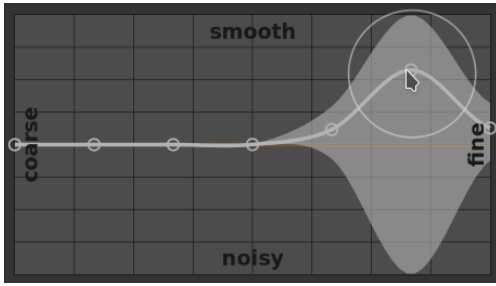
As discussed above, the *retouch* module allows you to choose how many detail levels to split your image into. It decomposes the image into separate layers and allows you to perform operations selectively on each individual layer or on the image as a whole:



See the [retouch](#) module documentation for more details.

### spline controls

The [denoise \(profiled\)](#), [raw denoise](#) and [contrast equalizer](#) modules allow their effects to be applied more or less to each wavelet scale using *splines*.



Here, each node in the graph represents a different level of detail in the image, from coarse detail on the left to fine detail on the right. You can raise or lower each of these nodes with your mouse to increase or decrease the module's effect, respectively, on that wavelet scale.

To modify the curve, click slightly above or below the line near to a node and drag up or down. You can change the width of your modification by scrolling with your mouse wheel, which increases or reduces the size of the circle displayed under your mouse pointer. A small circle indicates that the effect of dragging the curve up or down will be isolated mostly to the node being adjusted. A larger circle indicates that the effect will be more broad and will increasingly impact adjacent nodes. When you hover your mouse over the graph, shaded areas indicate the parts of the spline that will be impacted when you attempt to modify the curve.

### 3.4.7. deprecated modules

The darktable team regularly reviews old modules and updates their implementation where issues are found or updated science means that they can be improved. Most of the time we try to update existing modules with new functionality but occasionally that becomes problematic, often due to the overhead of having to maintain multiple versions of the module. In such circumstances a new replacement module is created and the old module becomes *deprecated*.

Module deprecation follows a fairly standard process: In the release that deprecates a module, that module can no longer be searched or added to custom module groups. It is moved into the read-only *deprecated modules* group and a message added to the module to warn of its removal and direct the user to a suitable alternative.

After a year, the module is removed from the *deprecated modules* group and is then only available for use in old edits (where that module was already active against a given image). Deprecated modules are never fully removed from darktable (in order to retain old edits) but development ceases and you are advised to use supported modules instead if you encounter an issue.

For example, modules deprecated in darktable 3.4 will be removed from the *deprecated modules* group in darktable 3.8 (assuming two major releases each year).

## 3.5. masking & blending

### 3.5.1. overview

Each processing module takes its input from the preceding module in the pixelpipe, performs its operation on the image data, and then hands the output to the next module in the pixelpipe.

A module's output data can optionally be reprocessed (combined) with its input data before being handed to the next module. This additional processing step is called *blending* - input and output data is reprocessed using algorithms called blending operators or [blend modes](#) .

Each blend mode is further controlled by the *opacity* parameter (having a value between 0% and 100%) which defines how much the input and output images contribute to the final result. Typically an opacity of 0% outputs an image which is identical to the input image (the module has no effect) whereas an opacity of 100% delivers the maximum effect of the module.

This opacity can be the same for every pixel (using the global opacity slider), in which case blending acts uniformly over the entire image. Alternatively the opacity values can vary depending on the properties or location of each pixel. This local modification of opacity is called a *mask*. Masks provide the user with fine control over which parts of an image are affected by a module and to what extent. The user may choose to activate a [drawn mask](#) , a [parametric mask](#) or a [combination](#) .

Blending and masking functionality is controlled via a group of icons located at the bottom of each applicable module.



These icons enable the following masking and blending options, from left to right:

#### off

Module output is passed to the next module in the pixelpipe without additional reprocessing. No further controls are displayed.

#### uniformly

Input and output images are reprocessed uniformly with the chosen blend mode, where the amount of blending is controlled by a single opacity slider. Additional controls are displayed to allow the blend mode and opacity to be selected. The default is a blend mode of "normal" with an opacity of 100%.

#### [drawn mask](#)

Reprocessing takes place with the chosen blend mode and an opacity based on pixel location as defined by one or more drawn shapes. Additional controls are displayed to allow mask elements to be drawn. If no mask elements are drawn then all pixels have the same opacity, as defined by the opacity slider.

#### [parametric mask](#)

Reprocessing takes place with the chosen blend mode and an opacity based on the properties of individual pixels. Additional controls are displayed to allow the opacity to be adjusted on a per-pixel basis, determined by pixel values.

#### [drawn & parametric mask](#)

Reprocessing takes place with the chosen blend mode and an opacity based on a combination of a drawn and parametric mask.

#### [raster mask](#)

Reprocessing takes place with the chosen blend mode and an opacity based on a mask that was generated by an active module earlier in the pixelpipe.

#### blending options

Choose which color space to use when calculating the blending mask, and specify whether or not to allow a mask to be generated based on the module's output channels (normally a parametric mask is generated based on the input channels coming into the module). The following options are available:

- *reset to default blend colorspace*: Use the default color space for the module to specify the parametric mask.
- *Lab*: Use the Lab color space (where available) to specify the parametric mask.
- *RGB (display)*: Use the display-referred RGB/HSL color space to specify the parametric mask.
- *RGB (scene)*: Use the scene-referred RGB/J<sub>z</sub>C<sub>z</sub>h<sub>z</sub> color space to specify the parametric mask.
- *show output channels*: Show the [parametric mask](#) output channel controls, so that the parametric mask can be defined in terms of the module's output channels.

---

**Note:** Not all of these blending options are available for every module.

---

## 3.5.2. blend modes


Blend modes define how the input and output of a module are combined (blended) together before the module's final output is passed to the next module in the pixelpipe.

Classic blending modes, designed for display-referred RGB (constrained to 0-100%), implicitly define a fulcrum at 50% (gray) or 100% (white) in their algorithms, depending on the blend mode. Because scene-referred is not subject to these restrictions, this fulcrum needs to be explicitly defined by the user when performing blending operations in the "RGB (scene)" color space. The additional *blend fulcrum* parameter will be presented to the user when using one of these blend modes in this color space. The effect depends on the operator used. For example, values above the fulcrum might be brightened and values below darkened, or vice versa.

The final output of a module is computed 'per-pixel' as follows:

$$\text{final\_output} = (1.0 - \text{opacity}) * \text{module\_input} + \text{opacity} * \text{blended\_output}$$

where the *blended\_output* is a combination of the input and output images, depending on the blend mode (below), and the *opacity* is defined 'per-pixel' by a combination of the mask and global opacity parameter. An opacity of 0% outputs an image that is identical to the input image of the module.

The "reverse" button  effectively reverses the roles of the input and output images in the 'per-pixel' computation:

$$\text{final\_output} = (1.0 - \text{opacity}) * \text{module\_output} + \text{opacity} * \text{blended\_input}$$

where the *blended\_input* is a combination of the output and input images, depending on the blend mode below where *output* and *input* image references are reversed. In "reversed" blend modes, an opacity of 0% outputs an image that is identical to the *output* image of the module.

## normal modes

### normal

The most commonly used blend mode, "normal" simply mixes input and output to an extent determined by the opacity parameter. This mode is commonly used to reduce the strength of a module's effect by reducing the opacity. This is also usually the blend mode of choice when applying a module's effect selectively with masks. This mode is also known as the "over" Porter-Duff alpha blending operator (see [alpha compositing](#) for more details).

### normal bounded

*not available in the "RGB (scene)" color space*

This blend mode is the same as "normal", except that the input and output data are clamped to a particular min/max value range. Out-of-range values are effectively blocked and are not passed to subsequent modules. Sometimes this helps to prevent artifacts. However, in most cases (e.g. highly color-saturated extreme highlights) it is better to let unbound values travel through the pixelpipe to be properly handled later. The "normal" blend mode is therefore usually preferred.

## arithmetic modes

### addition

Add together the pixel values of the input and output images, lightening the output. When blending in the "RGB (scene)" color space, the pixel values of the output image are multiplied by a value proportional to the "blend fulcrum".

### subtract

Subtract the pixel value of the *output* from the *input*. When blending in the "RGB (scene)" color space, the pixel values of the output image are multiplied by a value proportional to the "blend fulcrum". Pixel values less than 0 are set to 0.

**multiply**

Multiply the pixel values of the input and output together. When blending in display-referred color spaces, pixel values are between 0 and 1.0, the final output will be clamped and will always be darker. When blending in the “RGB (scene)” color space, this value is further multiplied by a value proportional to the “blend fulcrum”. In this case, values may be greater than 1.0 and therefore brighten the base image. This may have other side-effects, such as updating the white point in the filmic module.

Multiply blending simulates an optical variable density filter, where the density is defined by the output of the module. It has many applications, from blooming and local contrast enhancements (when used with a blur or low-pass filter) to dodging/burning and global contrast enhancements (when used with exposure). The fulcrum sets the output intensity threshold between darkening and brightening (any RGB value below fulcrum will darken).

**divide**

Divide the pixel values of the input by the output. When blending in the “RGB (scene)” color space, the pixel values of the output image are multiplied by a value proportional to the “blend fulcrum”.

Since this is the inverse of the multiply mode, it will darken where multiply brightens and vice versa. Everything else works in essentially the same way.

**screen**

*not available in the “RGB (scene)” color space*

Invert the input and output pixel values, multiply those values together and invert the result. This yields approximately the opposite effect to “multiply” mode – the resulting image is usually brighter, and sometimes “washed out” in appearance.

**average**

Return the arithmetic mean of the input and output pixel values.

**difference**

Return the absolute difference between the input and output pixel values.

**geometric mean**

Return the square root of the product of the input and output pixel values.

**harmonic mean**

Return the product of the input and output pixel values, multiplied by 2 and divided by their sum.

## contrast enhancing modes

The following modes are not available in the “RGB (scene)” blending color space as they rely on an assumption of “50% mid gray” which only applies to display-referred and non-linear color spaces.

**overlay**

This mode combines the “multiply” and “screen” blend modes: The parts of the input where the output is brighter, become brighter; The parts of the image where the output is darker, become darker; Mid-gray is unaffected. This mode is often combined with the [lowpass](#) and [highpass](#) modules.

**softlight**

This mode is similar to “overlay”, except the results are softer and less bright. As with overlay, it is often combined with the [lowpass](#) and [highpass](#) modules.

**hardlight**

This mode is not related to “softlight” in anything but name. Like overlay mode it is a combination of “multiply” and “screen” modes and has a different effect above and below mid-gray. The results with hardlight blend mode tend to be quite intense and usually need to be combined with a reduced opacity.

**vividlight**

This mode is an extreme version of overlay/softlight. Values darker than mid-gray are darkened; Values brighter than mid-gray are brightened. You will probably need to tone down its effect by reducing the opacity

**linearlight**

This mode is similar to the effect of “vividlight”.

**pinlight**

This mode performs a darken and lighten blending simultaneously, removing mid-tones. It can result in artifacts such as patches and blotches.

## color channel modes

### Lab channels

The following are available for blending in the Lab color space only

#### **Lab lightness**

Mix the lightness from the input and output images, while taking the color channels (a and b) unaltered from the input image. In contrast to “lightness” this blend mode does not involve any color space conversion and does not clamp any data. In some cases this blend mode is less prone to artifacts than “lightness”.

#### **Lab a-channel**

Mix the Lab “a” color channel from the input and output images, while taking the other channels unaltered from the input image.

#### **Lab b-channel**

Mix the Lab “b” color channel from the input and output images, while taking the other channels unaltered from the input image.

#### **Lab color**

Mix the Lab color channels (a and b) from the input and output images, while taking the lightness unaltered from the input image. In contrast to “color” this blend mode does not involve any color space conversion and does not clamp any data. In some cases this blend mode is less prone to artifacts than “color”.

### RGB channels

The following are available when blending in RGB color spaces only.

#### **RGB red channel**

Mix the “red” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “red” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

#### **RGB green channel**

Mix the “green” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “green” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

#### **RGB blue channel**

Mix the “blue” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “blue” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

### HSV channels

The following are available when blending in the “RGB (display)” color space only.

#### **HSV lightness**

Mix the lightness from the input and output images, while taking color unaltered from the input image. In contrast to “lightness” this blend mode does not involve clamping.

#### **HSV color**

Mix the color from the input and output images, while taking lightness unaltered from the input image. In contrast to “color” this blend mode does not involve clamping.

### others

#### **lightness**

Mix lightness from the input and output images, while taking color (chroma and hue) unaltered from the input image.

#### **chrominance**

Mix chrominance from the input and output images, while taking lightness and hue unaltered from the input image. This blend mode uses RGB ratios, divided by a Euclidean norm.

#### **lighten**

*not available in the “RGB (scene)” color space*

Compare the pixel values of the input and output images, and output the lighter value.

#### **darken**

*not available in the “RGB (scene)” color space*

Compare the pixel values of the input and output images, and output the darker value.

#### **hue**

*not available in the “RGB (scene)” color space*

Mix hue (color tint) from the input and output images, while taking lightness and chroma unaltered from the input image.

#### **color**

*not available in the “RGB (scene)” color space*

Mix color (chroma and hue) from the input and output images while taking lightness unaltered from the input image.

*Caution: When modules drastically modify hue (e.g. when generating complementary colors) this blend mode can result in strong color noise.*

#### **coloradjustment**

*not available in the “RGB (scene)” color space*

Some modules act predominantly on the tonal values of an image but also perform some color saturation adjustments (e.g. the [levels](#) and [tone curve](#) modules). This blend mode takes the lightness from the module’s output and mixes colors from input and output, enabling control over the module’s color adjustments.

## 3.5.3. masks

### 3.5.3.1. overview

Masks allow you to limit the effect of a module so that it only applies to certain parts of the image.

A mask can be regarded as a grayscale image where each pixel has a value between 0 and 1.0 (or between 0% and 100%). This value is the *opacity* and is used to determine how much a module affects each pixel.

The following sections explain how to construct masks in darktable.

### 3.5.3.2. drawn masks

With the drawn mask feature you can construct a mask by drawing shapes directly onto the image canvas. Shapes can be used alone or in combination. Once a shape has been drawn on an image it can be adjusted, removed, or reused in other modules.

Shapes are stored internally as vectors and are rendered with the required resolution during pixelpipe processing. Shapes are expressed in the coordinate system of the original image and are transformed along with the rest of the image by any active distorting modules in the pipe ( [lens correction](#) , [rotate and perspective](#) for example). This means that a shape will always work on the same image area regardless of any modifications that may be subsequently applied.

The controls required to create and alter drawn masks may be enabled by selecting either the “drawn mask” or “drawn & parametric mask” icon at the bottom of a module. You can also create and edit shapes using the [mask manager](#) module.

#### creating shapes

Choose a shape by clicking on the appropriate shape icon (from left to right: circle, ellipse, path, brush, gradient).



This will take you into the creation mode for that shape. Once you have finished drawing your shape you will automatically be taken into edit mode.

Ctrl+click on the shape icon to continuously draw multiple shapes of the same type – each time a shape is completed, you will re-enter creation mode for a new instance of that shape. While in continuous creation mode, right-click on the image to stop drawing shapes and enter edit mode.

For all drawn shapes, you can hold Shift while scrolling with the mouse wheel to change the extent of the shape's feathering (the blur at the edge of the shape) and use Ctrl+scroll to change the shape's opacity (how transparent it is). These operations are available in both creation and edit modes (as long as your mouse is over the shape in question).


By default, scrolling your mouse up increases the value of the relevant shape parameters. This behavior can be changed in [preferences > darkroom > scroll down to increase mask parameters](#) .

---

**Note:** When used in shape creation mode, the preceding scroll operations will also cause the *default* feathering or opacity to be changed. The new default values will be used the next time you create a new shape.

---

## editing shapes

Click on the 'show and edit mask elements' icon  to enter shape edit mode. This will show all drawn masks in use by the current module and allow you to edit those shapes. It will also expand the associated group in the [mask manager](#) . Ctrl+click on the same icon to enter restricted edit mode, where certain actions (e.g. dragging a complete shape or changing its size) are blocked. This is particularly useful to avoid costly mistakes when editing path and brush shapes.

Click and drag a shape to move it around the image canvas. Clicking on a shape will also select that shape in the mask manager.

## removing shapes

While in edit mode right-click on a shape to remove it.

## reusing shapes

You can reuse shapes that you have drawn in other modules. Click on the shapes drop-down (next to the 'show and edit mask elements' button) to choose previously-drawn shapes individually or to use the same group of shapes as used by another module. The following options are available for selection:

### add existing shape

Choose either an individual shape or a group of shapes that you've drawn previously (either within the [mask manager](#) or from within the drawn mask of another module). If that shape or group is used elsewhere, any changes you make will be reflected everywhere the shape or group is used.

### use same shapes as

Add a list of shapes used in another module to the current module's mask. This differs from the previous option in that it creates a new group of shapes, allowing shapes to be added to or removed from the group independently of the module from which they were copied. All shapes that are common to both groups remain linked.

## combining and managing shapes

The [mask manager](#) module can be used to manage your drawn shapes. This module also allows you to group and combine drawn masks using set operators (union, intersection, difference, exclusion).

## shape distortions

In order to ensure a consistent co-ordinate system, when you place a shape on the image, it is actually drawn on the original RAW file. This shape then passes up through the pixelpipe before finally being used by the module and drawn on the screen. This means that, if you have any enabled any distorting modules (such as lens correction), drawn shapes may appear distorted on the screen and in the final image. This can lead, for example, to circles being rendered as ellipses and gradient lines becoming curved. If you need to create a more accurate shape (to overcome these distortions) it is recommended that you avoid using the simple shapes (circles / ellipses) in favor of the path shape (which can be drawn using more points, reducing distortions). You can adjust the curve on gradient lines to overcome the simple distortions introduced by lens correction.



## available shapes

### circle

Click on the image canvas to place the circle. Scroll while hovering over the circle to change its diameter. Scroll while hovering over the circle's border to change the width of the feathering (the same effect as holding Shift while scrolling with the mouse wheel within the main shape).

### ellipse

The general principle is the same as for the circle shape. In addition, four nodes are shown on the ellipse line. Click on the nodes to adjust the ellipse's eccentricity. Ctrl+click and drag the nodes or use Shift+Ctrl+scroll (with the mouse wheel) to rotate the ellipse. Shift+click within the shape to toggle the gradual decay between equidistant and proportional mode.

### path

Click on the image canvas to place three or more nodes and generate a free-format enclosed shape. Terminate the path by right-clicking after having set the last point. By default, nodes are connected with smooth lines. If you want a node to instead define a sharp corner, you can do so by creating it with Ctrl+click.

In edit mode Ctrl+click on an existing node to convert it from smooth to sharp corners and vice versa. Ctrl+click on one of the line segments to insert an additional node. Right-click on a node to delete it. Take care to ensure that the mouse pointer is over the desired node and the node is highlighted, to avoid accidentally removing the whole path.

The size of the completed shape can be modified by scrolling. The same holds true for the width of the border (the area with a gradual opacity decay), which can also be changed with Shift+scroll (with the mouse wheel) from anywhere within the shape. Single nodes as well as path segments can be moved by dragging them with the mouse. If a node is selected by clicking on it, a further control point appears which allows you to modify the curvature of the line (reset to default by right-clicking). Dragging one of the control points on the border adjusts the border width just in that part of the shape.

Consider fine-tuning paths in restricted edit mode (enabled by Ctrl+clicking on the 'show and edit mask elements' icon). This allows you to adjust single nodes and segments without the risk of accidentally shifting or resizing the whole shape.

### brush

Start drawing a brush stroke by left-clicking on the image canvas and moving the mouse while keeping the button pressed. The brush stroke is finalized once you release the mouse button. Scroll the mouse to change the shape size, either before you start drawing or at any time during the operation. Likewise you can use the "{" and "}" keys to decrease/increase feathering, and the "<" and ">" keys to decrease/increase opacity.

If you have a graphics tablet with pen pressure sensitivity, darktable can apply the recorded pen pressure to certain attributes of the brush stroke. This operation can be controlled in [preferences > darkroom > pen pressure control for brush masks](#).

On lifting the tablet pen or releasing the left mouse button the brush stroke is converted into a number of connected nodes, which define the final shape. A configuration option ( [preferences > darkroom > smoothing of brush strokes](#) ) controls how much smoothing is applied. A higher level of smoothing leads to fewer nodes being created - this eases subsequent editing at the expense of lower accuracy.

Nodes and segments of a brush stroke can be modified individually. See the documentation on path shapes (above) for more details.

---

**Note:** Rendering a complex brush shape can consume a significant number of CPU cycles. Consider using the circle, ellipse or path shapes instead where possible.

---

### gradient

The gradient shape is a linear gradient which extends from a given point to the edge of the image.

Click on the image canvas to define the position of the line that defines 50% opacity. Dotted lines indicate the distance beyond which the opacity is 100% and 0%. Between these dotted lines the opacity changes linearly.

The line has two anchor nodes which you can drag to change the rotation of the gradient. You can also set the rotation angle when placing the gradient shape by clicking and dragging to place the shape.

Gradient lines can also be curved by scrolling with your mouse while hovering close to the center line. This can be useful to counteract the distortion caused by the [lens correction](#) module.

Depending on the module and the underlying image, using a gradient shape might provoke banding artifacts. You should consider activating the [dithering](#) module to alleviate this.

## reversing the polarity of a drawn mask

Click on the “+/-” button to reverse the polarity of the entire drawn mask. For example, a circular mask will, by default, cause the module to be applied only to the area inside the drawn circle. Reversing its polarity will cause the module to apply to the whole image, *except for* that circle.

### 3.5.3.3. parametric masks

The parametric mask feature offers fine-grained selective control over how individual pixels are masked. It does this by automatically generating an intermediate blend mask from user-defined parameters. These parameters are color coordinates rather than the geometrical coordinates used in drawn masks.

For each data channel of a module (e.g. Lab, RGB) and several virtual data channels (e.g. hue, saturation) you can construct a per-channel opacity function. Depending on each pixel’s value for a given data channel this function calculates a blending factor between 0 and 1 (100%) for that pixel.

Each pixel of an image thus has different blending factors for each of its data channels. All blending factors are finally multiplied together (pixel-by-pixel), along with the value of the global opacity slider, to form a complete parametric blend mask for the image.

If the blend mask has a value of 0 for a given pixel, the input of the module is left unchanged. If the blend mask has a value of 1 (100%) for a pixel, the module has its full effect.

#### channel tabs

Click on one of the channel tabs to select a data channel to use to build your mask.

Modules acting in (*display-referred*) Lab color space have data channels for L, a, b, C (chroma of LCh) and h (hue of LCh).

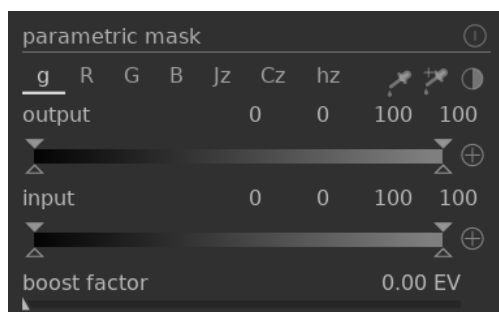
Modules acting in *display-referred* RGB color space have data channels for g (gray), R, G, B, H (hue of HSL), S (saturation of HSL), and L (lightness of HSL).

Modules acting in *scene-referred* RGB color space have data channels for g (gray), R, G, B, Jz (luminance component of JzCzhz), Cz (chroma, or saturation, of JzCzhz), and hz (hue of JzCzhz). The g (gray) value is calculated as a weighted average of the R, G & B channels, the exact weightings depending on the working color space being used. The JzCzhz color space is a polar representation of the Jzazbz color space, in the same way that LCh is a polar representation of the Lab space. Like the L in Lab color space, the Jz is a representation of the luminosity of a pixel that aligns with how we perceive brightness. However, the Jzazbz color space is much better for high dynamic range images and is less susceptible to hue shifts than Lab space.

See [Wikipedia](#) for more details about these color spaces.

Two sliders can be shown for each associated data channel: one that works on the *input data* that the module receives and one that works on the *output data* that the module produces prior to blending. The sliders for the output data channels are hidden by default and can be shown using the *show output channels* option in the blending menu.

The *boost factor* slider allows the range of values targeted by the parametric mask sliders to be extended. It may be used in scene referred editing, where luminance values may extend beyond 100%, to target highlights. This slider is only available for channels where it is meaningful.



## inspecting data channels & masks

Press the letter C while hovering over a channel's input/output slider to view the input/output image data for that color channel. The center image changes to display that color channel either in gray-scale values or in false colors depending on the setting in [preferences > darkroom > display of individual color channels](#) .

Press the letter M to see the resulting mask for that slider overlaid on the image.

When the mouse pointer leaves the slider the image returns to normal after a short delay.

## linear / log mode

Press the letter A while hovering over the a slider to change its display to 'log' mode. This provides more fine control in the shadows. Press A again to toggle back to 'linear' mode.

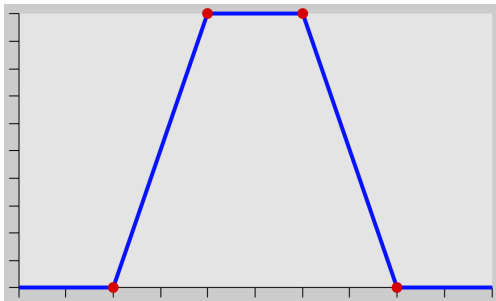
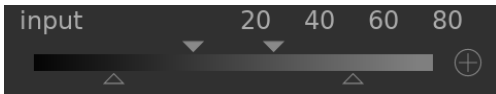
## channel input/output sliders

With each color channel slider you can construct a trapezoidal opacity function. For this purpose there are four markers per slider. Two filled triangles above the slider mark the range of values where opacity is 1. Two open triangles below the slider mark the range values where opacity is 0. Intermediate points between full and zero are given a proportional opacity.

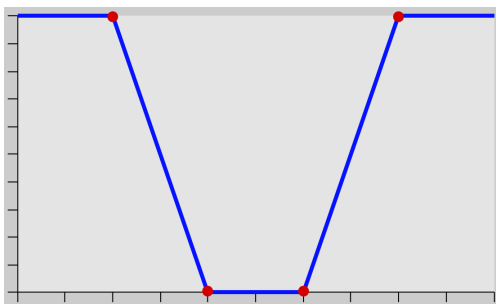
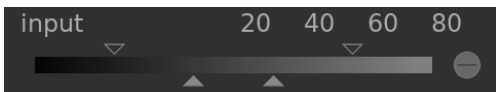
The filled triangles, or inside markers, indicate the closed (mostly narrower) edge of the trapezoidal function. The open triangles, or outside markers, indicate the open (mostly wider) edge of the trapezoidal function. The sequence of the markers always remains unchanged: they can touch one another but they cannot switch position.

A polarity (+/-) button to the right of each the slider switches between "range select" and "range de-select" modes, with visual confirmation provided by exchanging the upper and lower triangle markers. These two types of trapezoidal functions are represented graphically in the following images.

### range select



### range deselect



In their default state all markers are at their extreme positions.

In this state a range select function selects the whole range of values giving an “all at 100%” mask. Starting from there one can move the sliders inwards to gradually exclude more and more parts of the image except for the remaining narrow range.

Conversely a range de-select function (enabled by toggling the polarity) by default deselects the whole range of values, giving an “all-zero” mask as a starting point. Moving the sliders inwards gradually includes more and more parts of the image except for the remaining narrow range.

### color pickers

With the left-hand color picker button you can select a point or area probe from your image. The corresponding values for the real and virtual data channels are then displayed within each color channel slider.

With the right-hand color picker button you can automatically set the slider’s values based on the selected range. Click and drag to set the parameters for the input slider from the drawn rectangle; Ctrl+click and drag to set the parameters for the output slider.

### invert

Click the invert button above the sliders to invert the polarity of the entire parametric mask. This differs from the polarity buttons beside the individual sliders which just invert the parameters for the current slider/channel.

### reset

Click the reset button above the sliders to revert all parametric mask parameters to their default state.

## 3.5.3.4. combining drawn & parametric masks

Drawn and parametric masks can be used in combination to form a single mask that can be applied to a module.

There are two main elements which control how individual masks are combined: the *polarity* setting of each individual mask (defined by the plus or minus buttons) and the setting in the “combine masks” combobox.

The “combine masks” combobox contains the following options, defining how the drawn and parametric masks will be combined:

#### **exclusive**

A straightforward method of combining masks, exclusive mode multiplies together the individual pixel values from each of the component masks.

For a given pixel, the final mask will have value of 0 if *any* of the individual masks are 0 at that location and it will only have a value of 1.0 if *all* masks have a value of 1.0 at that location.

Any individual mask can *exclude* a pixel by setting its value to 0, regardless of what the other masks do. Once a pixel is excluded by a mask there is no way for another mask to include it again.

#### **inclusive**

Inclusive mode first inverts each individual mask (subtracts its value from 1.0), multiplies the inverted masks together, and finally inverts the combined mask once again.

For a given pixel, the final mask will have a value of 1.0 if *any* of the individual masks are 1.0 at that location and it will only have a value of 0 if *all* masks have a value of 0 at that location.

Any individual mask can *include* a pixel by setting its value to 1.0, regardless of what the other masks do. Once a pixel is fully included by a mask (its value is 1.0) there is no way for another mask to exclude it again.

---

### exclusive and inclusive inverted modes

Using the above combination methods alone would still be rather limiting. We gain maximum flexibility by allowing an additional inversion step for each individual mask. This is governed by the polarity buttons that you find close to the individual channels.

Toggling the polarity button of a mask inverts its values (subtracts the original value from 1.0).

Finally within the “combine masks” combobox you may select the *exclusive & inverted* or *inclusive & inverted* options. Each of these options is equivalent to the *exclusive* and *inclusive* modes, respectively, but with a final step that inverts the resulting mask.

### typical use cases

#### inclusive mode

For this mode you set the “combine masks” combobox to inclusive mode and make sure that all polarity buttons of all the individual channels and of the drawn mask are set to negative (-). Your starting point is a mask where all pixels have a value of zero (no pixel is selected). You now adjust the parametric mask sliders to bring more and more pixels into the selection or you draw shapes on the canvas to select specific areas of your image.

#### exclusive mode

In the opposite case you set the “combine masks” combobox to exclusive mode and make sure that all polarity buttons are set to positive (+). Your starting point is a mask with all values at 1.0 (all pixels are selected). You now adjust the parametric mask sliders to exclude parts of your image as needed or directly draw shapes on the canvas to exclude those areas.

For your convenience the parametric masks GUI provides a toggle button that inverts all channel polarities and toggles between inclusive and exclusive mode in the “combine masks” combobox.

For novice users it is recommended that you stick to the above two use cases. This means that you should decide beforehand how you want to construct your mask.

### 3.5.3.5. mask refinement & additional controls

When a parametric or drawn mask is active, several additional sliders are shown which allow the mask to be further refined.

#### details threshold

This control allows you to alter the opacity of the mask based on the amount of detail in the image. Use this slider to select either areas with lots of detail (positive values) or areas that are flat and lacking in detail (negative values). The default (zero) effectively bypasses details refinement. This is mostly useful to apply sharpening and blurring effects that ignore out-of-focus (bokeh) regions or to sharpen only blurry parts, preventing over-sharpening of in-focus regions.

---

**Note:** The data used for the detail mask refinement is taken from the demosaic stage in the processing pipeline, and not from the module’s input (which is used for the other parametric mask criteria). None of the processing modules after demosaic will have any effect on the detail mask and it is not currently available for non-RAW images.

---

#### feathering guide

Mask feathering smooths a drawn or parametric mask such that the mask’s edges automatically align with the edges of features in the image. The smoothing is guided either by the module’s input or output (before blending), and may happen before or after the mask blurring, depending on what is selected in the “feathering guide” combobox. Feathering is particularly sensitive to the choice of guide image when used with edge-modifying modules (modules for sharpening or blurring an image).

- *output before blur*: feathering is guided using the *output* image of the module and takes place *before* the mask is blurred
- *input before blur*: feathering is guided using the *input* image of the module and takes place *before* the mask is blurred
- *output after blur*: feathering is guided using the *output* image of the module and takes place *after* the mask has been blurred
- *input after blur*: feathering is guided using the *input* image of the module and takes place *after* the mask has been blurred

**feathering radius**

Adjust the strength of the feathering effect. Feathering works best if the mask's edges already approximately match some edges in the guiding image. The larger the "feathering radius" the better the feathering algorithm can align the mask to more distant edges. If this radius is too large, however, the feathered mask may overshoot (cover regions that the user wants to exclude). Feathering is disabled when the feathering radius is set to 0.

**blurring radius**

Blurring the mask creates a softer transition between blended and unblended parts of an image and can be used to avoid artifacts. The mask blur slider controls the radius of a gaussian blur applied to the final blend mask. The higher the radius, the stronger the blur (set to 0 for an unblurred mask). Mask blur is always applied after feathering if both kinds of mask adjustment are activated. This allows any resulting sharp edges or artifacts to be smoothed.

**mask opacity**

The strength of the module's effect is determined by the mask's local opacity. Feathering and blurring the mask may reduce the opacity of the original mask. The "mask opacity" slider allows you to readjust the mask opacity to compensate. If the mask opacity is decreased (negative slider values) less opaque parts are affected more strongly. Conversely, if the mask opacity is increased (positive slider values) more opaque parts are affected more strongly. As a consequence, completely opaque portions of the mask always remain opaque and completely transparent portions always remain transparent. This is to ensure that regions that have been fully excluded from or included in a module's effect (by setting the mask's opacity to 0% or 100%) remain fully excluded or included.

**mask contrast**

This slider increases or decreases the mask contrast. This allows you to adjust the transition between the opaque and transparent parts of the mask.

**temporarily switch off mask**

Sometimes it is useful to visualize a module's effect without the mask being active. Click on this icon to temporarily deactivate the mask (the selected blend mode and global opacity remain in effect).

**display mask**

Click on this icon to display the current mask as a yellow overlay over a black-and-white version of the image. Solid yellow indicates an opacity of 100%; a fully visible gray background image (without yellow overlay) indicates an opacity of 0%.

example: feathering a drawn mask



It can be rather tedious to create a drawn mask that precisely covers a particular feature in an image. In this example, we want to enhance the color contrast of the lion sculpture shown in the left image above without affecting the background.

1. The first image above shows the original, unaltered image.
2. The second image shows a rough selection of the sculpture created with a drawn mask. Note that the mask is rather fuzzy and does not precisely follow the outline of the lion sculpture.
3. The third image shows the effect of adjusting the feathering radius, mask opacity and mask contrast, leading to a well matched mask with little effort. In this example the feathering radius has been adjusted to 50 and a blur radius of 5 was chosen to smooth the mask to some degree. The mask opacity and mask contrast have been increased to 0.3 and 0.5, respectively.
4. The final image above shows the end result, where the color enhancement (via the [color contrast](#) module) is restricted to only the lion sculpture.

Mask feathering works particularly well in this example because the sculpture is well separated from the out-of-focus background. The distinct edge at the border of the sculpture guides the feathering mask adjustment to match the shape of the sculpture.

### 3.5.3.6. raster masks

As described in the previous sections, the final output of a module's mask (the combined effect of any drawn and parametric masks) is a grayscale raster image representing the extent to which the module's effect should be applied to each pixel. This raster image is stored internally for active modules and can be subsequently reused by other modules in the pixelpipe.

As with any mask, if the opacity value for a pixel in a raster mask is zero the module's input passed through the module unchanged. If the opacity is 1.0 the module has its full effect. For each value between 0 and 1.0 the module's effect is applied proportionally at that location.

You can choose a raster mask from the combobox. Raster masks can be identified by the name of the module against which they were originally generated.

---

**Note:** Raster masks are generated as part of a module's internal processing. Once a module's processing is complete its mask then becomes available to subsequent modules in the pixelpipe.

This has two implications:

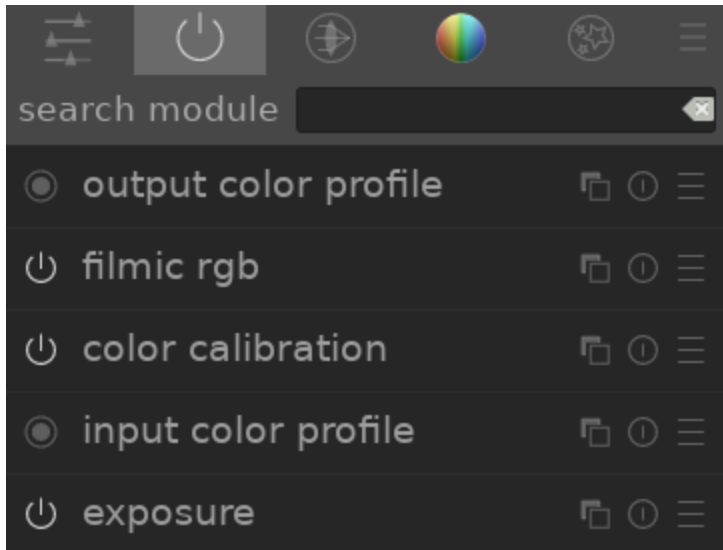
1. Raster masks cannot be generated by disabled modules since they do not participate in pixelpipe processing. As soon as you disable a module, its mask is no longer available for use.
2. Raster masks are passed *up* the pixelpipe after module processing - they can only be used by modules that come later in the pipe than the generating module.

---

## 3.6. organization

### 3.6.1. overview

Processing modules are organized and accessed via the right-hand panel in the darkroom:



Click on the icons at the top of this panel to reveal, from left to right,

#### **quick access panel**

A customizable panel allowing quick access to commonly-used module controls.

#### **active modules**

A group containing the modules that are currently active in the pixelpipe. Right-click to show all modules that are present in the [history stack](#) within the active group, regardless of whether or not they are actually active.

#### **module groups**

One or more groups of processing modules. These groups are user-defined but some default groupings are provided as presets.

#### **presets menu**

A menu that allows you to access stored module layout presets and create your own (via the “manage presets..” option). You can also directly access the “manage presets” dialog by Ctrl+clicking on the presets menu.

Click once on a module group icon (including the active group) to show only the modules in that group. Click a second time to show a list of all modules that are currently active or present in any group.

You can change which widgets appear in the quick access panel and which modules appear in the module groups, by right-clicking the appropriate icon.

## search

Below the module group icons is the search bar, which you can use to access any of the processing modules, regardless of whether they are currently in a group. Some modules have additional tags stored against them that will also be queried when searching.

### 3.6.2. module groups

A number of pre-defined module groups are shipped with darktable and are selectable as presets. These are summarized below.

All of these presets (with the exception of *modules: deprecated* and *search only*) also include the quick access panel. All except the *modules: deprecated* group include the search bar.

#### *modules: all*

This preset contains all modules, sorted according to the traditional module groupings used prior to darktable 3.4, as follows:




**base modules**

The minimal set of modules normally required to render a presentable image.


**tone modules**

Other modules related to tone levels and contrast.


**color modules**

Modules related to color grading and color profiles.


**corrective modules**

Modules related to correcting problems such as lens distortions, sensor noise, sharpening, etc.


**(special) effects modules**

“Special effect” modules such as retouch, liquify, bloom, sharpen, etc.

*modules: default*

This preset is the default module group layout from darktable 3.4 onwards and consists of a simplified set of module groups, as follows:


**technical modules**

Modules that deal with technical issues relating to the physics of sensors and denoising, lenses and associated corrections, color profiles, dynamic range and tone mapping, and recovering from damage to the image by physical limitations (hot pixels, clipped highlights, etc.).


**grading modules**

Modules concerned with primary (corrective) and secondary (creative) corrections of colors and tones.


**(special) effects modules**

“Special effect” modules such as retouch, liquify, bloom, sharpen, etc.

*workflow: scene-referred & workflow: display-referred*

These presets define groups of modules relevant to the [scene-referred and display-referred workflows](#), sorted into groups as shown below:


**base modules**


A basic set of modules to adjust the cropping/orientation, adjust the exposure, and apply tone mappings and contrast as appropriate to the workflow.


**color modules**

Modules relating to color grading and color saturation.


**corrective modules**


Modules relating to correcting problems relating to lens distortions, sensor noise, sharpening, retouching, etc.


**(special) effects modules**

“Special effect” modules such as watermark, framing, vignetting, etc.

*workflow: beginner*

This preset provides a minimal set of modules targeted as a starting point for beginners. It is suggested that beginners start by copying this minimal preset, and add to it as they gain experience with other modules.


**base modules**

A basic set of modules to adjust the cropping/orientation, adjust the exposure, and apply a basic tone mapping via base curve or basic adjustments.

 **grading modules**

Modules dealing with creative tone and color grading.

 **(special) effects modules**

“Special effect” modules such as retouch, sharpen, watermark, etc.

*previous config*

These presets are automatically generated for users who have upgraded from a version of darktable prior to 3.4. Where you have previously set up favourites or altered the *hidden* flag on modules, these presets contains those customisations, retaining the legacy module groups (*previous config* preset) or new module groups (*previous config with new layout* preset).

If favourites were created in prior versions these will remain available in an additional group:

 **favourite modules**

This group was previously used by users to make it easier to find frequently-used modules, and is available under the “previous config” presets. New users can, of course, still create their own custom group and name it “favourites” if they so desire.

*search only*

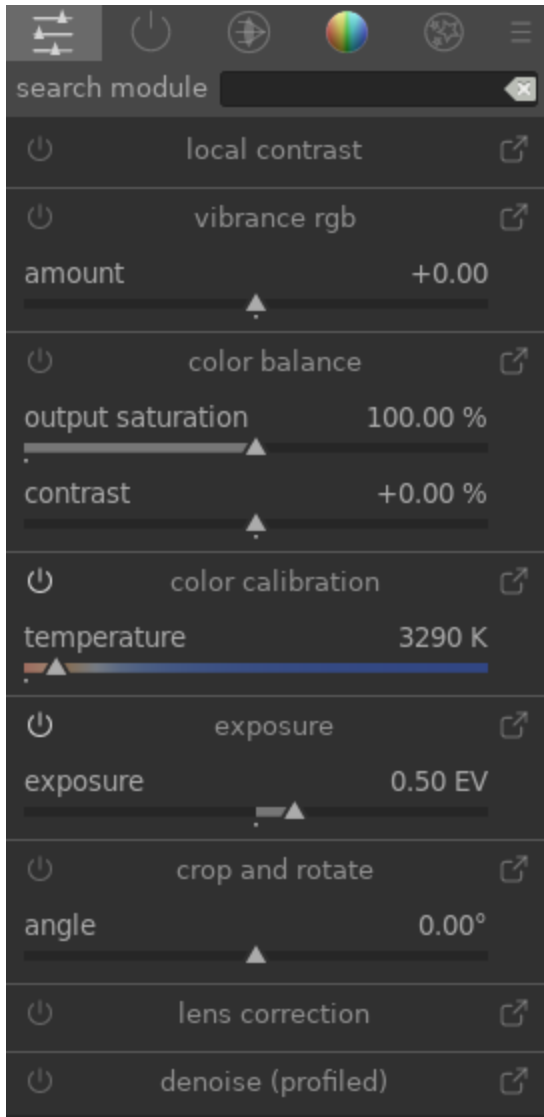
This preset does not include any module groupings. Modules may only be accessed using the search facility.

*modules: deprecated*

This preset contains a list of deprecated modules. This is the only way to access deprecated modules for new edits but be warned: these modules will be removed for new edits in the next release of darktable. This group cannot be duplicated and the modules within it cannot be added to user-created groups.

### 3.6.3. quick access panel

The quick access panel allows you to access widgets from a number of processing modules all in one place.



You can add new widgets by right-clicking on the icon at the top-left or by using the [manage module layouts](#) screen.

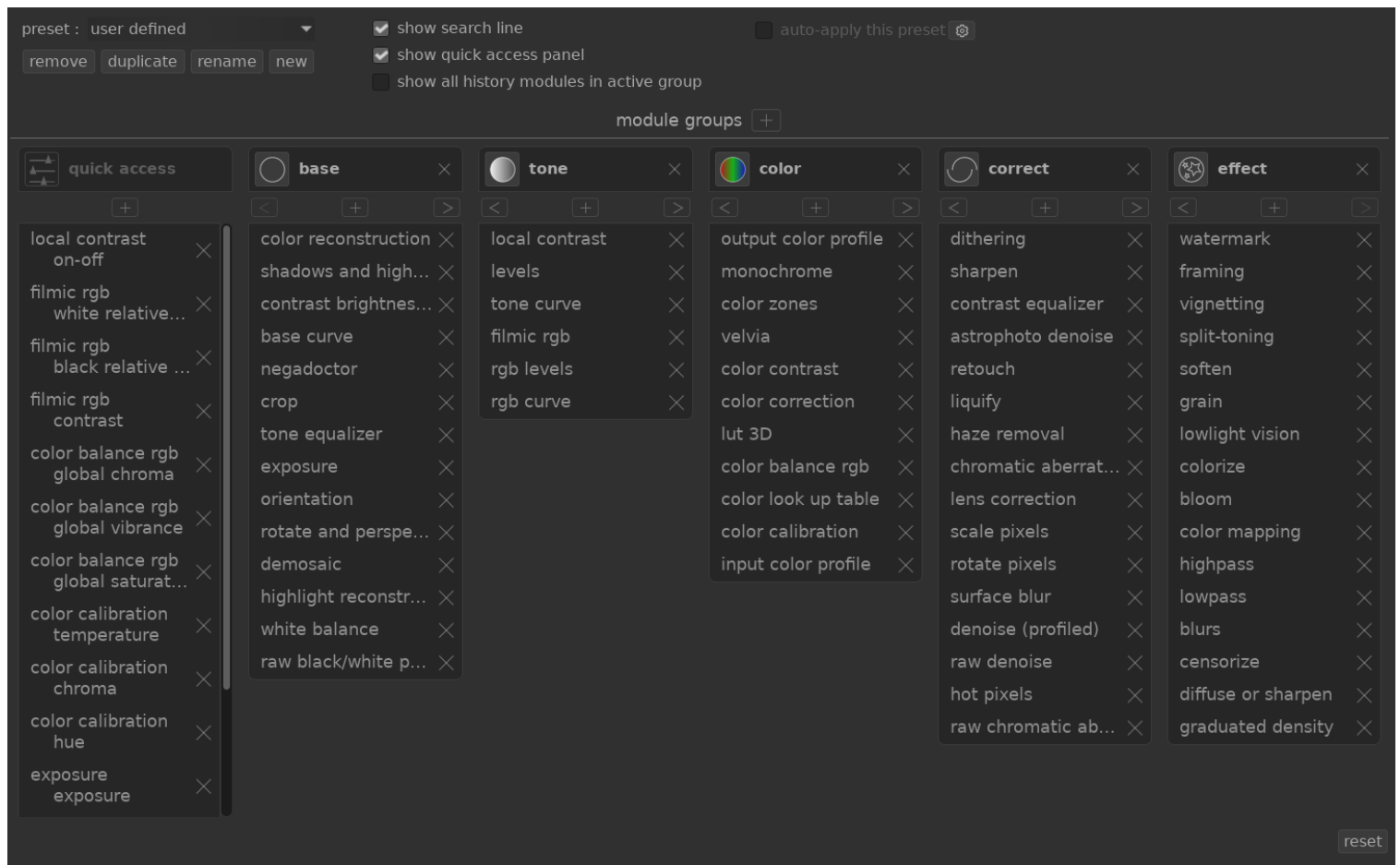
Click on the icon to the right of the module name to open the full version of that module. Click on the icon to the left of the module name to enable/disable the module.

Currently only some widgets may be added to this panel (check boxes, sliders, comboboxes) - more complex widgets such as curves are not supported.

If any module has [multiple instances](#) enabled you will no longer be able to manage that module's widgets through the quick access panel and will have to use the full module interface.

### 3.6.4. manage module layouts

Manage the layout and grouping of processing modules and the quick access panel.



This maintenance screen can be accessed from the *presets* menu beside the module search box or module group icons (below the scopes module in the darkroom view). Ctrl+click on the preset menu to open this screen directly.

Settings are automatically saved when you exit the screen. Click *reset* to abandon any changes made in the current editing session.

## module controls

### global controls

The following global controls are available in the top panel of the screen.

#### preset

Select an existing module group preset.

#### remove

Remove the current preset (user-defined presets only).

#### duplicate

Duplicate the current preset with a new name. The above example shows a new preset named “user defined”, which has been created by duplicating the “modules: default” preset.

#### rename

Rename the current preset (user-defined presets only). Right-click to bring up a menu that can be used to copy, paste, select all, delete or insert an emoji).

#### new

Create a new preset containing a minimal list of modules.

#### show search line

Choose whether to display the search bar below the module group icons

**show quick access panel**

Choose whether to display the quick access panel. If checked, a new entry will appear in the bottom panel to allow you to add or remove widgets.

**show all history modules in active group**

Select this to show all modules that are present in the [history stack](#) within the active group, regardless of whether or not they are actually active.

**auto-apply this preset**

Module group presets can be automatically applied based on the type of image being worked on. The check box indicates whether this preset currently has any auto-apply rules. Click on the gear icon to amend the auto-apply settings. See [presets](#) for details.

## module groups

The bottom panel of the screen allows you to alter the quick access panel and module groups for the selected preset (user-defined presets only).

**add a group**

Click on the + sign beside the “module groups” label to add a new group.

**remove a group**

Remove a group by clicking on the X button beside the group name.

**add a module/widget**

Add a module to a group, or a widget to the quick access panel, by clicking the + sign below the group name. Select the required module/widget from the displayed list.

**remove a module/widget**

Remove a module from a group, or a widget from the quick access panel, by clicking the X beside the module/widget name.

**change a group’s icon**

Change the icon assigned to a group by clicking on the existing group icon and selecting a new one from the displayed list.

**change the group order**

Change the order in which the groups are displayed by clicking on the < and > buttons below the group names.

**rename a group**

Rename a module group by clicking on the group name and typing. Right-click to bring up a menu that can be used to copy, paste, select all, delete or insert an emoji)

## 4. Tethering

### 4.1. overview

The tethering view allows you to capture images directly into darktable from a connected camera.

To use the tethering feature you first need to connect your camera to your PC using a USB cable. Your computer might ask to mount or view the connected camera. *Do not mount or view the camera.* If your camera is mounted or viewed automatically, you will need to “unmount/eject” the camera before darktable can access it. This unlocks the camera so that darktable can take control of it – darktable will then re-lock the camera so that it cannot be used by other applications.

After the USB cable is connected, go to the [import](#) module in the [lighttable](#) view. If your camera is available for use, a new section should appear in the import module containing the name of your camera and a “mount camera” button. Click this button to connect your camera and three additional buttons will appear: “copy & import from camera”, “tethered shoot” and “unmount camera”. Click “tethered shoot” to enter the tethering view. When you are finished, press the “unmount camera” button before physically disconnecting it.

darktable uses [gphoto2](#) to interface with your camera. If you have problems finding the connected camera as described above, check the [troubleshooting](#) section in this chapter to verify that your camera has tethering support.

In the center view, images are shown while you capture them. You can capture an image by either using darktable’s user interface or by manually triggering a capture with your camera. If you are using Live View the image will be shown in darktable’s center view.

When entering tethering view, a film roll will be created using the same structure as defined for camera import (see [preferences > import > session options](#) ). The job code will be predefined as “capture”.

If you want to group your captures into different film rolls, you should use the [session](#) module in the right-hand panel to set a different job code. When entering a new name and pressing Enter, a new film roll will be created and newly-captured images will be added into this new film roll.

darktable provides some useful tools to set up an image capture in the user interface. You can set up timelapse captures, brackets for HDR and even sequential captures of bracketed images. For more information read the documentation on the [camera settings](#) module and the [examples](#) later in this chapter.

### 4.2. tethering view layout

#### left panel

##### [image information](#)

Display image information.

#### right panel

From top to bottom:

##### [scopes](#)

A graphical depiction of the current image’s light levels and colors. This module can be moved to the left-hand panel if desired (see [preferences > miscellaneous > position of the scopes module](#) ).

##### [session](#)

Session settings.

##### [live view](#)

Live view settings.

##### [camera settings](#)

Camera settings.

### [metadata editor](#)

Edit metadata for selected images.

### [tagging](#)

Tag selected images.

## bottom panel

From left to right:

### [star ratings](#)

Apply star ratings to images.

### [color labels](#)

Apply color categories to images.

## 4.3. examples

The following sections show two typical use cases for the tethering view.

### studio setup with screening

This is a pretty common use case. You have your studio and subject set up, the camera is connected to your computer and tethering view is active in darktable. You work at the camera and take images. Whenever you like, you can review the images directly on your computer monitor instead of using the camera LCD.

This workflow is efficient and effective, since you can immediately review your captures instead of waiting until after the shoot when everyone is gone. If you're shooting with a model this is a nice way to preview the captures with the client instead of fumbling around with your camera.

Working in the tethering view can save you time and aggravation. Set a [session](#) name, shoot your images, and they will be saved to the correct film roll for the session, for easy on-site review.

### capturing a timelapse

A timelapse is a video clip composed of images taken in a timed sequence. A typical example is to take a timelapse of a cityscape to capture the movement of the clouds and traffic etc.

To set up a timelapse capture, create a new [session](#) . Now decide whether you want to shoot in manual or auto mode. It is recommended that you only use auto mode in situations where the ambient light will change significantly during the time of the shoot, for example, when shooting a timelapse over 24 hours.

The [camera settings](#) module can be used to define the delay (the number of seconds between capture) and sequence (the number of images to capture) of your timelapse.

To start the timelapse click the capture button in the same panel and watch the [filmstrip](#) fill up with images. The last captured image is always displayed in the center view.

## 4.4. troubleshooting

This troubleshooting guide can be used to verify whether or not your camera can be used with tethering. This is done using the same ( [gphoto2](#) ) tool that darktable uses to interface with your camera.

Before you start, you first need to find your camera port name. Usually the port "usb:" is sufficient and is therefore used in the following guide.

## is your camera detected?

The following command will verify that your camera is connected to the computer and detected by gphoto2.

```
env LANG=C gphoto2 --auto-detect
```

## check the camera's driver abilities

Execute the following command and ensure that the “capture choices” ability supports “Image” and configuration support is “yes” - darktable will check these two abilities to decide whether to show the “tethered shoot” button.

```
env LANG=C gphoto2 --port usb: --abilities
```

## remote capture

This step will verify that your camera can be remotely controlled - i.e. that it can capture an image, download it to your computer and display it within darktable.

```
env LANG=C gphoto2 --port usb: --capture-image-and-download
```

## tethered capture

This last step tests if your camera supports “events”, are heavily used by darktable. Running this command will make the gphoto2 process wait for an image capture event which you must manually trigger on your camera. If successful, the image will be downloaded to your computer.

```
env LANG=C gphoto2 --port usb: --capture-tethered
```

## now what?

If any of the above steps failed, there are problems with your specific camera and driver. Please file an issue on the [gphoto2 github page](#) . Add the following flags to the failed command for better support and attach the log output to your issue:

```
--debug --debug-file gphoto2_debug.log
```

If you successfully completed all of the above tests, your camera is probably supported by darktable. If these tests were successful but you nevertheless stumble upon a problem in darktable, please file an issue on the [darktable github page](#) . Please attach the log output from the commands above and the log file output produced by starting darktable with the following command:

```
darktable -d camctl 2>1 >camctl.log
```



# 5. Map

## 5.1. overview

The map view allows you to see where your geo-tagged images were taken, and to add location information to non-geo-tagged images.

The map view shows a world map with the images in the current [collection](#) pinned to their geo-tagged location (if available). This requires that images are tagged with location information. Some newer cameras, including smartphones, are already equipped with GPS receivers. Other cameras may need additional GPS hardware to do this.

Even if your camera doesn't support this feature, there is an alternative method – darktable can match the Exif date/time in your image(s) to a separate GPX data tracking file created by a GPS tracker, that has recorded your movements. GPS trackers can be purchased as standalone handheld devices or you can install a GPS tracker app on your smartphone. Location tagging with GPS tracking data can be done using the [geotagging](#) module, in the lighttable and map views.

## center map view

In the center of the map view you will see a world map.

Map data is taken from open map sources on the internet. As such, new map data is only available if you are connected to the internet – darktable keeps a disk cache of previously loaded map data.

You can navigate within the map using your mouse. Left-click and drag to move the map. Use the scroll-wheel to zoom in and out.

On-screen controls and displays are available to help you find your way. A navigation area is located on top left of the map – use this as an alternative to mouse-dragging and scrolling. The scale of the map is displayed at the bottom-left. At the bottom-right you can see the geographical coordinates for the center of the map.

Images that already have geo-location attributes in their metadata are displayed as small icons on the map. Images close to each other are grouped and a count of grouped images is displayed on the bottom-left corner.

In order to assign geo coordinates to an image, activate the [filmstrip](#) on the lower panel (press Ctrl+F). You can assign a geo location to an image by dragging the image icon from the film-strip and positioning it on the map – darktable will record the new location (latitude and longitude ) as part of the image metadata. This data will be included in exported images.

In order to remove location data from an image simply drag it from the map and drop it onto the filmstrip.

Images close to each other are grouped under a single image group. You can use the [map settings](#) module to control the grouping as needed. The number displayed on the bottom left of the thumbnail is the number of images inside the group. A white number means that all images in the group are at exactly the same location, whereas a yellow number means they are not. Use the mouse scroll wheel while hovering over a group of images to scroll through the thumbnails of the images in that group.

Normally, images in the center map view have black borders. If an image is selected in the filmstrip, then the corresponding image on the map will be highlighted with a white border.

Click+drag to adjust the location of an image. Shift+click to move a complete group of images.

The left and right panels provide additional controls (see [map view layout](#) ).

## undo/redo

All image movements in the map view are recorded by darktable. It is possible to undo or redo such changes to recover a previous state. Note that this undo/redo facility is unlimited while moving images but it is reset each time you leave the map view.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

## 5.2. map view layout

### left panel

These modules are identical to the lighttable view. From top to bottom:

#### **[collections](#)**

Filter the list of images displayed in the map view.

#### **[recently used collections](#)**

View recently used collections of images.

#### **[image information](#)**

Display image information.

### right panel

Here you can find the modules specific to the map view. From top to bottom:

#### **[find location](#)**

Search for a place on map.

#### **[locations](#)**

Manage a hierarchical list of location tags and their corresponding regions on the map.

#### **[map settings](#)**

Choose the map provider and set up various map display parameters.

#### **[tagging](#)**

Tag selected images.

#### **[geotagging](#)**

Apply GPX track data to selected images.

### bottom panel

#### **[filmstrip](#)**

Drag images from the filmstrip onto the map as described in the map [overview](#) .

# 6. Slideshow

## 6.1. overview

The slideshow view allows you to watch a slideshow of your current collection with the associated filtering rules and sort order applied.

To learn more about how to define the collection and filtering rules, see the section on [collections](#) . Select the sort criteria and sort order of the images in the [top panel](#) .

The next section provides more details on the [usage](#) of the slideshow view.

## 6.2. usage

The slideshow view is still in an early stage of development with only a basic set of features.

If you don't need the auto-advance mode, you could even use the [sticky preview feature](#) instead.

spacebar	start and stop auto-advance mode which automatically switches to the next images every five seconds by default.
ESC	leave slideshow mode and return to lighttable view.
+ or up arrow	increase delay between each image.
- or down arrow	decrease delay between each image.
left-click or right arrow or right shift-key	switch to the next image of the collection.
right-click or left arrow or left shift-key	switch to the previous image of the collection.

---

**Hint:** To take full advantage of your screen size, put darktable into fullscreen mode by pressing F11 and hide the border-controls by pressing the B key.

---

# 7. Print

## 7.1. overview

This view allows you to print your images. Because printing is not easy, there are many technical aspects to be taken into account.

After selecting an image in the [lighttable view](#) you can enter the [print settings](#) module to adjust printer settings and initiate printing.

This module supports the printer's ICC profile, which is somewhat mandatory if you want to obtain a high quality print close to the image displayed on the screen.

It is important to note that ICC profiles provided by the paper and/or printer manufacturers cannot be used on GNU/Linux as they are printer-driver dependent. The darktable print module uses CUPS and there are no ready-to-use ICC profiles available for this driver.

## 7.2. print view layout

The central area displays the image layout on the paper (the white area). Some gray borders may be displayed around the image to represent the printable area (the page minus the borders) not filled by the image.

The [filmstrip](#) below the image allows you to select more images.

### left panel

#### [collections](#)

Filter the list of images displayed in the lighttable.

#### [image information](#)

Display image information

### right panel

#### [print settings](#)

Adjust print settings and initiate printing.

# 8. Module Reference

## 8.1. overview

The modules in this reference section are broken down into two distinct types:

### processing modules

Processing modules are used exclusively in the darkroom view. Each module performs a processing operation on the image before passing its output to the next module for further processing. Together this sequence of processing steps forms the [pixelpipe](#).

### utility modules

Utility modules may be used in any darktable view. They are not directly involved in processing the pixels of an image but perform other ancillary functions (managing image metadata and tags, editing history, modifying pixel pipeline order, snapshots and duplicates, image export etc.).

The two types of modules have a few aspects in common, described below.

## module header

Each module has a header at the top, normally consisting of the following elements:

### module name

Click on the name to expand or collapse the rest of the module and show/hide its controls.

### reset parameters button

This normally appears to the right of the module name and is used to reset the state of the module back to its original condition.

### presets menu

This normally appears at the far right of the module header. The [presets](#) menu is predominantly used in processing modules, but many of the utility modules allow presets to be defined as well. You can also access this menu by right-clicking anywhere on the module header.

Processing modules contain additional elements in their module header, as described in the [processing module header](#) section.

## module resizing

## utility modules

Some utility modules contain lists of information that can grow as more entries are added. To help manage screen real-estate, it is possible to increase or decrease the maximum number of entries that can be displayed before a scroll bar is added. To alter the maximum number of entries, place the mouse over an entry in the list, and hold Ctrl while scrolling your mouse wheel. If the list currently contains more entries than this maximum, a scrollbar will appear so that you can access the hidden entries.

---

**Note:** It is not possible to extend these areas beyond the number of entries currently shown. If you attempt to do so using Ctrl+scroll, the maximum number of entries *will* increase, and a toast message will appear informing you of the new maximum. However, the module itself will not be resized unless its content exceeds this maximum.

---

## processing modules

Some processing modules contain drawn graphical elements that can take up too much or too little screen space depending on the width of your side panels. These drawing areas usually default to a 16:9 aspect ratio and can be similarly resized by hovering over them and holding Ctrl while scrolling.

## 8.2. processing modules

### 8.2.1. astrophoto denoise

Remove image noise while preserving structure.

This is accomplished by averaging each pixel with some surrounding pixels in the image. The weight of such a pixel in the averaging process depends on the similarity of its neighborhood with the neighborhood of the pixel being denoised. A patch with a defined size is used to measure that similarity.

As denoising is a resource-intensive process, it slows down pixelpipe processing significantly. Consider activating this module late in your workflow.

### module controls

#### patch size

The radius of the patch used for similarity evaluation.

#### strength

The strength of the denoising.

#### luma

The amount of denoising to apply to luma. Select carefully in order not to lose too much structure.

#### chroma

The amount of denoising to apply to chroma. You can be much more aggressive with this parameter.

### 8.2.2. base curve

Simulate the in-camera JPEG by applying a characteristic base curve to the image.

darktable comes with a number of base curve presets that attempt to mimic the curves of various camera manufacturers. These presets are automatically applied according to the manufacturer ID found in the image's Exif data. Camera-specific base curve presets are also available for some camera models.

This module will be enabled by default if [preferences > processing > auto-apply pixel workflow defaults](#) is set to "display-referred". A second option in the preferences dialog allows you to choose whether darktable should attempt to apply a camera-specific base curve (if found) or the generic manufacturer one.

### module controls

Please refer to the [curves](#) section for more details about how to modify curves including the **scale for graph** and **preserve colors** controls.

#### fusion

Trigger the exposure fusion feature (see [this Wikipedia article](#) ). This feature allows you to merge the image with one or two copies of itself after applying the current base curve and boosting its exposure by a selected number of EV units. The resulting image is thus a combination of two or three different exposures of the original image.

Use this feature to compress the dynamic range of extremely underexposed images or for true HDR input. For best results, use the [exposure](#) module to apply a suitable adjustment for correctly exposed highlights.

#### exposure shift (fusion)

The exposure difference between the merged images in EV units (default 1). This slider is only visible if the *exposure fusion* feature is activated.

### exposure bias (fusion)

Determines how the multiple exposures are computed. With a bias of 1 (the default), the image is fused with overexposed copies of itself. With a bias of -1, it is fused with underexposed copies. A bias of 0 attempts to preserve the overall lightness of the image by combining both over- and under-exposed copies of the image. This slider is only visible if the *exposure fusion* feature is activated.

## 8.2.3. basic adjustments (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.6 and should no longer be used for new edits. Please use the [quick access panel](#) instead.**

---

A convenience module that combines controls from [exposure](#) , [highlight reconstruction](#) , [color balance](#) and [vibrance](#) into a single module.

While this module can provide a quick and convenient way to make simple adjustments to an image, it must be used with care. Normally exposure adjustments should come before [input color profile](#) in the pixelpipe, and color adjustments should come after. Because the *basic adjustments* module combines all of these functions into a single operation in the pixelpipe, it may not play nicely with other modules. Therefore, if you plan to use *basic adjustments* with other modules, please instead consider using the [exposure](#) + [base curve](#) / [tone curve](#) / [filmic rgb](#) + [color balance](#) modules so that these operations occur in the correct places in the pixelpipe.

## module controls

### black level correction

Equivalent to black level correction in the [exposure](#) module, this slider defines the threshold at which dark gray values are cut off to pure black. Reducing this can bring some dark colors back into gamut. Increasing this slider may appear to increase the contrast and pop of an image, but it can push dark colors out of gamut, and clipped data cannot be recovered further down the pixel pipe. To control the contrast in the shadows, it is better to use other modules such as [tone curve](#) or [levels](#) , which mitigate these negative impacts further down the pixel pipeline.

### exposure

Adjust exposure compensation. Adding 1 EV of exposure compensation is equivalent to doubling the exposure time in camera, opening the aperture by 1 stop, or doubling the ISO.

Positive exposure corrections will make the image brighter. As a side effect noise level is increased. Depending on the basic noise level of your camera and the ISO value of your image, positive exposure compensations with up to 1EV or 2EV should still give reasonable results.

Negative exposure corrections will make the image darker. Given the nature of digital images this can not correct for fully blown out highlights but allows you to reconstruct data if only some of the RGB channels are clipped (see the [highlight reconstruction](#) module for information on how to deal with clipped pixels).

### highlight compression

Compress the highlights of the image in order to recover detail.

### contrast

Equivalent to the *contrast* slider in the [color balance](#) module, this is used to increase the separation of luminance values around a fulcrum point, in effect making the *tone curve* steeper. The *middle gray* slider sets the fulcrum point for the contrast, and any pixels whose luminance matches this middle gray point will be unaffected. Pixels brighter than the middle gray point will become even brighter, and pixels darker than the middle gray point will become even darker.

### preserve colors

If a non-linear tone curve is applied to each of the RGB channels individually, then the amount of tone adjustment applied to each color channel may be different, and this can cause hue shifts. Therefore, the *preserve colors* menu provides different methods of calculating the “luminance level” of a pixel. The amount of tone adjustment is calculated based on this luminance value, and then this same adjustment is applied to all three of the RGB channels. Different luminance estimators can affect the contrast in different parts of the image, depending on the specific characteristics of that image. The user can therefore choose a particular estimator that provides the best results for the given image.

### middle gray

Set the fulcrum point for the *contrast* slider. This is equivalent to the *contrast fulcrum* slider on the [color balance](#) module. If the contrast slider is set to 0 this slider will not have any effect.

**brightness**

Equivalent to increasing the *gamma factor* slider in the [color balance](#) module. Moving the slider to the right will increase the brightness of the image, with an emphasis on the mid-tones.

**saturation**

Equivalent to the *output saturation* slider on the [color balance](#) module, this slider affects the saturation, or brilliance and intensity of the colors. Moving the slider to the right will increase the amount of color in the image; moving the slider to the left will reduce the amount of color in the image.

**vibrance**

Accentuate the colors of the image without adding unnatural colors, as it's often the case with the *saturation* slider. It works by reducing the lightness of already saturated pixels to make the colors more vivid. You can also achieve some interesting effects by combining it with the saturation slider to target more or less saturated areas of the image.

**auto**

Automatically adjust the exposure, taking into account the entire image, or use the color picker to select a rectangular area of the image - the exposure will be automatically adjusted based on the selected region. This allows you to prioritise which parts of the image should be well-exposed.

**clip**

This affects the number of pixels that will be clipped to black or white during the auto-exposure calculation. Moving this slider to the right will allow more pixels to be clipped and increase the contrast; moving this slider to the left will compress the image more and lower the contrast.

## 8.2.4. bloom

Create a soft bloom effect.

This module works by blurring the highlights and then blending the result with the original image.

---

**Note:** This module performs blurs in Lab color space, and is no longer recommended. Instead, use the [tone equalizer](#) module or the [exposure](#) module with a [parametric mask](#).

---

## module controls

**size**

The spacial extent of the bloom effect.

**threshold**

The threshold for the brightness increase.

**strength**

The strength of the overlighting.

## 8.2.5. blurs

Simulate physically-accurate blurs in scene-referred RGB space.

## blur types

Three types of blur are provided:

1. *lens blur*: Simulates a lens diaphragm with a configurable number of blades and blade curvature to create synthetic bokeh.
2. *motion blur*: Simulates the effect of camera motion with a configurable path.
3. *gaussian blur*: This is not really an optical blur but can be used for denoising or for creative effects using [blend modes](#)

A diagram at the top of the module shows the shape of the blurring operator (known as the [point spread function](#)). The module will turn each luminous point from the scene into a blot shaped like the displayed blurring operator, with the size of the blot defined by the *blur radius*.



## module controls

### general

#### blur radius

The spreading size of the blur.

#### blur type

Choose between the different blur variants (above).

### controls specific to lens blur

#### diaphragm blades

The number of blades that the diaphragm is composed of. Older lenses used typically 5 or 7 blades, newer lenses typically use 9 or 11 blades. In any case, real lenses have an odd number of blades and any number greater than 11 blades comes very close to producing a perfect disc. If you degenerate the diaphragm settings with the *concavity* to create a star or an asterisk, this control defines how many branches it has.

#### concavity

- a concavity of 1 ensures the diaphragm is a [regular convex polygon](#) (triangle, pentagon, heptagon, etc.).
- a concavity greater than 1 but lower than number of blades - 1 turns the shape into a star.
- a concavity greater than number of blades - 1 but lower than number of blades turns the shape into an asterisk, when decreasing *linearity* below 1.
- a concavity greater than or equal to number of blades degenerates the shape into a “burst pattern”.

#### linearity

- a linearity of 0 creates a disc, no matter the number of blades or the concavity.
- a linearity of 1 makes all the outer bounds of the shape straight.
- a linearity between 0 and 1 makes the outer bounds of the shape more or less curved.

#### rotation

Allows the shape to be rotated with respect to its center – mostly useful with a small number of blades, when a particular orientation is needed.

### controls specific to motion blur

#### direction

The orientation of the motion’s path in angular degrees. 0° is horizontal motion.

#### curvature

The curvature of the motion. Zero produces a straight line, a negative value produces a concave curvature, a positive value produces a convex curvature.

#### offset

Shifts along the motion path following its curve. This is useful to select a portion of the curved path that is symmetrical, which produces a coma shape (example 1: direction = -45°, curvature = +2, offset = +0.5 ; example 2 : direction = -45°, curvature = +1, offset = +1).

### caveats

This module is implemented using a “naive” convolution, which is a slow algorithm. Faster approaches are available (using FFT) but not yet implemented. The GPU implementation, through OpenCL, should hide this issue somewhat. In any case, the runtime of the module will increase with the square of the blur radius.

The blurring process does not take scene depth and depth-of-field into account, but blurs the whole image as a flat object. It is therefore not suitable for creating fake depth-of-field. Using darktable’s general masking will only partially work to isolate the foreground of an image, since it will still be blurred into the background.

### tips and tricks

All images are usually (even a tiny bit) noisy. If you blur only a part of the image, the blurred region will look suspiciously clean compared to the rest of the image. It is therefore a good idea to add a bit of noise on top of the blurred part to blend it with the rest, using either the [grain](#) or the [sensorize](#) modules.

## 8.2.6. censorize

Degrade parts of the image in an aesthetically pleasing way, in order to anonymize people/objects or hide body parts.

Censorize works in linear RGB color space to apply a physically-accurate gaussian blur and gaussian luminance noise.

Aside from anonymization, this module can also be used for a wide range of creative purposes, for example:

- Combine a simple blur with a multiply [blend mode](#) to create a realistic bloom (Orton effect).
- Combine a simple blur with a subtract blending mode and low opacity to create an [unsharp mask](#), similar to the [sharpen](#) module but in an RGB scene-referred space.
- Add noise to create artificial grain.

---

**Note:** The anonymizing methods provided by this module are not forensically safe in order to favor aesthetics. Some forensic techniques may still be able to reconstruct the censored content based on its structure, especially for simple shapes and text (e.g. license plates, street numbers).

If forensically safe anonymization is required, the only way to achieve this is to paint the surfaces with a solid color.

**The darktable team does not accept responsibility for poorly anonymized pictures leading to the identification of individuals or personal property.**

---

### workflow

You are advised to leave the module's controls at their default values while you [mask](#) the areas of the image that you wish to censorize, in order that the details of the image remain visible.

### module controls

#### input blur radius

The strength of the first pass of the gaussian blur.

#### pixellation radius

The size of the "big pixels" created after the first pass of gaussian blur.

#### output blur radius

The strength of the second pass of the gaussian blur, applied after the pixellation.

#### noise level

The strength (standard deviation) of the luminance gaussian noise applied after the second pass of the gaussian blur. Adding noise can fake details in the blurred regions and make content detection more difficult for artificial intelligence algorithms.

## 8.2.7. channel mixer (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [color calibration](#) module instead.**

---

A simple yet powerful tool to manage color channels.

This module accepts red, green and blue channels as an input and provides red, green, blue, gray, hue, saturation and lightness channels as output. It allows you to independently control how much each input channel contributes to each output channel.

### RGB matrix multiplication

You can think of the *channel mixer* as a type of matrix multiplication between a 3x3 matrix and the input [R G B] values.

$$\begin{bmatrix} R\_out \\ G\_out \\ B\_out \end{bmatrix} = \begin{bmatrix} R_r & R_g & R_b \\ G_r & G_g & G_b \\ B_r & B_g & B_b \end{bmatrix} \times \begin{bmatrix} R\_in \\ G\_in \\ B\_in \end{bmatrix}$$

If, for example, you've been provided with a matrix to transform from one color space to another, you can enter the matrix coefficients into the *channel mixer* as follows:

- set the destination to *red* then set the R<sub>r</sub>, R<sub>g</sub> & R<sub>b</sub> values using the red, green and blue sliders
- set the destination to *green* then set the G<sub>r</sub>, G<sub>g</sub> & G<sub>b</sub> values using the red, green and blue sliders
- set the destination to *blue* then set the B<sub>r</sub>, B<sub>g</sub> & B<sub>b</sub> values using the red, green and blue sliders

By default, *channel mixer* just copies the input [R G B] channels straight over to the matching output channels. This is equivalent to multiplying by the identity matrix:

$$\begin{bmatrix} R\_out \\ G\_out \\ B\_out \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} R\_in \\ G\_in \\ B\_in \end{bmatrix}$$

As an example use case, the following matrix is useful for taming ugly out-of-gamut blue LED lights by making them more magenta:

$$\begin{bmatrix} 1.00 & -0.18 & 0.18 \\ -0.20 & 1.00 & 0.20 \\ 0.05 & -0.05 & 1.00 \end{bmatrix}$$

In this case it is useful to use a [parameteric mask](#) to limit the effect of the *channel mixer* to just the problematic colors.

A more intuitive take for what the *channel mixer* sliders do:

- for the *red* destination, adjusting sliders to the right will make the R, G or B areas of the image more red. Moving the slider to the left will make those areas more cyan.
- for the *green* destination, adjusting sliders to the right will make the R, G or B areas of the image more green. Moving the slider to the left will make those areas more magenta.
- for the *blue* destination, adjusting sliders to the right will make the R, G or B areas of the image more blue. Moving the slider to the left will make those areas more yellow.

## monochrome

Another very useful application of the *channel mixer* is the ability to mix the channels together to produce a grayscale output – a monochrome image. Use the *gray* destination, and set the red, green and blue sliders to control how much each channel contributes to the brightness of the output. This is equivalent to the following matrix multiplication:

$$\text{GRAY\_out} = [r \ g \ b] \times \begin{bmatrix} R\_in \\ G\_in \\ B\_in \end{bmatrix}$$

When dealing with skin tones, the relative weights of the three channels will affect the level of detail in the image. Placing more weight on red (e.g. [0.9, 0.3, -0.3]) will make for smooth skin tones, whereas emphasising green (e.g. [0.4, 0.75, -0.15]) will bring out more detail. In both cases the blue channel is reduced to avoid emphasising unwanted skin texture.

Different types of traditional black and white film have differing sensitivities to red, green and blue colors, and this can be simulated by setting the gray destination coefficients appropriately. The *channel mixer* module has a number of built-in presets that can be used to achieve this.

## module controls

### destination

Select the destination channel that will be affected by the slider settings immediately below. The red, green and blue destination channels are used for color mixing as described in the *matrix multiplication* section above. The gray channel is used for making grayscale images as described in the *monochrome* section above. It is also possible to define the R, G & B input channels to produce HSL (hue, saturation and lightness) values on the output, although this is a very specialised application.

### red

Defines how much the red input channel should contribute to the selected destination channel.

**green**

Defines how much the green input channel should contribute to the selected destination channel.

**blue**

Defines how much the blue input channel should contribute to the selected destination channel.

## 8.2.8. chromatic aberrations

Correct chromatic aberrations.

In contrast to the [raw chromatic aberrations](#) module, this module does not require raw data as input.

### workflow

To obtain the best result, you are advised to proceed as follows:

1. Attenuate the chromatic aberrations as much as possible in the [lens correction](#) module using the TCA sliders.
2. Increase the strength slider in this module to better see its effect.
3. Increase the radius until the chromatic aberrations disappear. If this is insufficient, try enabling the “very large chromatic aberrations” setting.
4. Choose the guide that gives the best result in term of sharpness and artifacts.
5. Reduce the strength to avoid washing out the colors too much.

For more complicated cases you could also try the following:

- Use several instances with different correction modes – for example, a first instance in “brighten only” mode, and a second in “darken only” mode.
- Use several instances with low strength to correct the chromatic aberrations a little at a time without degrading colors too much.
- Use the module with parametric or drawn masks.
- Use RGB red, green and blue blend modes to restrict the effect to a particular channel.

### module controls

**guide**

The color channel that will be used as a reference for the correction.

**radius**

The radius of the effect. Increase until chromatic aberrations are eliminated. This is the most important slider of the module.

**strength**

This slider acts as a safeguard and can help to preserve colorful areas that do not suffer from chromatic aberrations. Increase for stronger correction, decrease for stronger preservation.

**correction mode**

Allows you to restrict the effect to brighten or darken pixels only. For full control, this can be used in combination with R, G, B blend modes and multiples instances.

**very large chromatic aberrations**

Makes the algorithm iterative to help in reducing very large chromatic aberrations.

## 8.2.9. color balance

A versatile tool for adjusting the image’s color balance.

This module can be used to revert parasitic color casts or to enhance the visual atmosphere of an image using color grading, a popular technique in the cinema industry. For scene-referred workflow, you should consider using the improved [color balance rgb](#) module instead.

### overview

The *color balance* module allows you to shift colors selectively by luminance range (shadows, mid-tones, and highlights). It can do this using two different methods:

**lift, gamma, gain**

The classic method, which allows a more separated control of shadows versus highlights.

**slope, offset, power**

The new standard defined by the American Society of Cinematographers Color Decision List (ASC CDL) and more suited to scene-referred editing.

The *master* settings affect the whole image. They are not available in *lift, gamma, gain (sRGB)* mode. The slider ranges are limited to usual values ([50%; 150%] for saturation, [-50%; 50%] for contrast), but higher and lower values can be defined via keyboard input after right-clicking on the corresponding slider.

For better efficiency, in *slope, offset, power* mode it is recommended that you set the slope first, then the offset, and finally the power, in that order. The name of the mode can be use as a mnemonic to remember the order.

The shadows parameter has a far heavier effect in *slope, offset, power* mode than in *lift, gamma, gain* mode. When switching from the former to the latter, you should adapt the saturation in shadows, dividing by around 10.

---

**Note:** Although this module acts on RGB colors its location in the pixelpipe puts it into the Lab color space. Accordingly the module converts from Lab to RGB, performs its color adjustments, and then converts back to Lab.

---

## presets

Several presets are provided in this module to help you to better understand how it can best be used. The “teal/orange color-grading” preset is a very popular look in cinema and is a good showcase model. It is meant to be used with two instances combined with [masks](#) . The first instance will exclude skin tones and will shift neutral colors toward teal blue. The second will partially revert the first one and add more vibrance to skin tones only. Together they will create separation between subject and background. The masking and blending parameters will need to be tweaked to suit every image.

Other presets provide Kodak film emulations. In this way you can recreate any film look you like using *color balance*.

## module controls

**mode**

*lift, gamma, gain (sRGB)* is the legacy mode from darktable 2.4 and earlier. In this mode, the color transformations are applied in sRGB color space encoded with the sRGB gamma (average gamma of 2.2).

*lift, gamma, gain (ProPhoto RGB)* is the same as the previous mode but works in ProPhoto RGB space, encoded linearly. In this mode, the RGB parameters are corrected in XYZ luminance (Y channel) internally so they affect only the color, and only the “factors” adjust the luminance.

*slope, offset, power (ProPhoto RGB)* applies the ASC CDL in ProPhoto RGB space, encoded linearly. As with the previous mode, the RGB parameters are corrected in XYZ luminance internally. In this mode, the slope parameter acts as an exposure compensation, the offset acts as a black level correction, and the power acts as a gamma correction. All parameters will have some impact on the whole luminance range but the slope will mostly affect the highlights, the offset will mostly affect the shadows, and the power will mostly affect the mid-tones.

**color control sliders**

This combobox selection affects the user interface used for the shadows, mid-tones and highlights controls.

*RGBL* controls allow direct access to the RGB parameters that will be sent to the algorithm and internally adjusted in XYZ luminance, depending on the mode used. They are the only ones stored in darktable’s development history.

*HSL* controls are more intuitive, but are only an interface: the hues and saturations are computed dynamically from and to the RGB parameters and never stored. During the HSL to RGB conversion, the HSL lightness is always assumed to be 50%, so the RGB parameters are always balanced to avoid lightness changes. However, during the RGB to HSL conversion, the HSL lightness is not corrected.

As a consequence, editing in RGB, then in HSL, then again in RGB will not retain the original RGB parameters, but will normalize them so their HSL lightness is 50%. The difference is barely noticeable in most cases, especially using the modes that already correct the RGB parameters internally in XYZ luminance.

In both modes, additional “factor” sliders act on all RGB channels at once. Their effect is similar to the controls of the [levels](#) module and affect only the luminance.

---

### input saturation

A saturation correction applied before the color balance. This can be used to dampen colors before adjusting the balance, to make difficult images easier to process. When you entirely desaturate the image, this creates a luminance-based monochrome image that can be used as a luminance mask, to create color filters with the *color balance* settings, like a split-toning or sepia effect (when used with blending modes).

### output saturation

A saturation correction applied after the color balance. This is useful once you have found a proper hue balance but find the effect too heavy, so you can adjust the global saturation at once instead of editing each channel saturation separately at the expense of possibly messing up the colors.

### contrast / contrast fulcrum

The contrast slider allows the luminance separation to be increased. The fulcrum value defines the luminance value that will not be affected by the contrast correction, so the contrast will roll over the fulcrum. Luminance values above the fulcrum will be amplified almost linearly. Luminance values below the fulcrum value will be compressed with a power function (creating a toe). This correction comes after the output saturation and is applied on all RGB channels separately, so hues and saturations might not be preserved in case of dramatic settings (shadows might be resaturated, highlights might be desaturated, and some color shift is to be expected).

### shadows, mid-tones, highlights

Depending on the mode used, the shadows settings will control either the lift or the offset, the mid-tones settings will control either the gamma or the power, and the highlights settings will control either the gain or the slope. Parameters are transferred unaltered when you change the mode.

In *RGBL* mode, the RGB sliders' range is limited to [-0.5; 0.5]. In *HSL* mode, the saturation sliders range is limited to [0%; 25%]. Values outside of these bounds can be defined with keyboard input by right-clicking on the slider.

---

**Note:** The shadows, mid-tones and highlights sliders can take up a great deal of space in the *color balance* module. The overall layout of these sliders can therefore be cycled through three different layouts by clicking on the *shadows*, *mid-tones*, *highlights* heading. You can also set the default layout with [preferences > darkroom > colorbalance slider block layout](#) .

---

### optimize luma

The color-picker beside the optimize luma label will select the whole image and optimize the factors for shadows, mid-tones and highlights so that the average luminance of the image is 50% Lab, the maximum is 100% and the minimum is 0%, at the output of this module. This is essentially histogram normalization, similar to that performed by the [levels](#) module. The optimizer is only really accurate when used in *slope*, *offset*, *power* mode.

If you want more control, you can define three control patches by using the color pickers beside each factor slider to sample luminance in selected areas. The *shadows* color picker samples the minimum luminance, the *mid-tones* color picker samples the average luminance, and the *highlights* color picker samples the maximum luminance. The most sensitive parameter is the mid-tones factor, since selecting a slightly different area can lead to dramatic parameter changes. Using the factors color pickers alone, without triggering the luma optimization, will allow you to perform adjustments without general optimization, but each parameter is always computed taking the other two into account. Once patches are selected, the label changes to read "optimize luma from patches". To reset one patch, you can just redo the selection. Patches are not saved in the parameters and are retained only during the current session.

It is important to note that the luminance adjustment targets only the output of the *color balance* module and does not account for adjustments performed in other modules later in the pipeline (e.g. [filmic rgb](#) , [tone curve](#) , [color zones](#) , [levels](#) ). Using the *color balance* module to remap the luminance globally on the image is not recommended because it does not preserve the original colors - modules such as [tone curve](#) or [filmic rgb](#) are better suited for this purpose. Luminance adjustments in *color balance* are better performed, in combination with color adjustments, using masks.

## neutralize colors

In an image where some areas are exposed to direct sunlight and some areas are exposed to reflected light (shadows), or where several artificial light sources are present simultaneously, shadows and highlights often have different color temperatures. These images are particularly difficult to correct since no general white balance will match all the colors at once. The color neutralization optimizer aims at helping you find the complementary color for shadows, mid-tones, and highlights so that all the color casts are reverted, and the average color of the image is a neutral gray.

As with the luma optimization, the color picker beside the neutralize colors label will trigger a general optimization over the whole image. This works fairly well in landscape photography, or for any photograph with a full spectrum of colors and luminances.

For night and events photography, this will most likely fail and you will need to manually input the sampling areas with the color-pickers beside each hue slider. For the highlights sample, use a color exposed to spotlights that should be neutral white or light gray. For the shadows sample, use a color exposed to ambient light that should be neutral black or dark gray. For the mid-tones sample, use a color exposed by both ambient and spotlights.

The success of the optimization depends on the quality of the samples. Not every set of samples will converge to a good solution and you need to ensure that the color patches you choose are really a neutral color in real life. In many cases the optimizer will output the correct hue but an excessive saturation that will need some extra tweaking. In some cases, no valid optimization will be delivered and you will need to reset the saturation parameters and start over, or simply stop after the patches selection. Note that in the auto-optimization, the maximum saturation is 25%, which might not be enough in very few cases but will avoid inconsistent results in most.

If you select color patches from the hue color pickers without triggering the optimization, the software will only perform one round of optimization and then stop. This allows you to control each luminance range separately and avoid divergence of the solution in corner cases. The hue and saturation corrections are computed taking into account the two other luminance ranges and three factors, and will always output the complementary color of the selected area. If you want to reinforce the color of the area instead, you can then add 180° to the computed hue. Once patches are selected, the label changes to read "neutralize colors from patches". To reset one patch you can just redo the selection. Patches are not saved in the parameters and are retained only during the current session. The parameters found by the automatic neutralization are accurate only in *slope*, *offset*, *power* mode, but can work to some extent in *lift*, *gamma*, *gain* mode too.

## 8.2.10. color balance rgb

An advanced module which brings color-grading tools from cinematography into the photographic scene-referred pipeline.

This module is not suitable for beginners with no prior knowledge of color theory, who might want to stick to the *global chroma* and *global vibrance* settings until they have a good understanding of the [dimensions of color](#).

### introduction

Color-grading is an important part of image editing. It can help to remove unwanted color casts and can also deliver a creative color twist that will add atmosphere to your images. In the days of film photography, most of the color ambience was obtained with the film emulsion and the developing chemicals, with some color timing being performed under the enlarger with color heads. This consumed expensive resources and was mostly reserved for the cinema industry, where the job was performed by a colorist.

In the digital age, where raw images look flat and even, color-grading assumes the same role that film emulsions did, by re-introducing color shifts for aesthetic purposes. It can also serve to harmonize the color palette of a series of images (which may have been shot under different conditions) to achieve a consistent global look. For this task, the [vectorscope](#) is also extremely useful.

Colorists usually split color-grading into two distinct steps:

1. *Primary color-grading* aims to fix unwanted color casts and create a neutral starting point,
2. *Secondary color-grading* gives the image its final look and atmosphere.

Primary color-grading is best left to the [color calibration](#) module, which operates in a physical framework better suited to illuminant correction. *Color balance RGB*, on the other hand, is mostly concerned with secondary color-grading. Performing a truly neutral primary color-grading should make the secondary color-grading easy to transfer between images (via styles, presets or copy & paste) with a similar effect.



## general principles

The *color balance RGB* module is an improvement over the [American Society of Cinematographers Color Decision List](#) (ASC CDL), and uses alpha masks to allow the effect to be properly split between shadows and highlights. The classic CDL acts on the entire luminance range, and each of its parameters is given more weight on some parts of the image only as a side-effect of the mathematics.

This module works, for the most part (*4 ways, chroma, vibrance, contrast*), in a linear RGB color space designed specifically for color-grading. This color space exhibits a uniform spacing of perceptual hues while retaining a physically-scaled luminance<sup>1</sup>. The perceptual part of the module (*saturation* and *brilliance*) works in the JzAzBz<sup>2</sup> color space, which provides a perceptual scaling of both lightness and chromaticity suitable for HDR images. Both color spaces ensure that saturation and chroma changes take place at constant hue, which is not the case for most other saturation operators in darktable (notably in the older [color balance](#) module).

The *color balance RGB* module expects a scene-referred linear input and produces a scene-referred RGB output, which may or may not be linear, depending on the module settings (*contrast* and *power* will delinearize the output).

At its output, *color balance RGB* checks that the graded colors fit inside the pipeline RGB color space (Rec 2020 by default) and applies a soft saturation clipping at constant hue, aiming to retarget out-of-gamut color to the nearest in-gamut color by scaling both chroma and lightness. This prevents the chroma and saturation settings from pushing colors outside of the valid range and allows more drastic adjustments to be safely used.

*Note that this module abides by the CIE definitions of chroma and saturation, as explained in the [dimensions of color](#) section.*

## module controls

### master tab

#### hue shift

Rotate all colors in the image by an angle over the chromaticity plane, at constant luminance and chroma. You can use this control to remove spilled colored light on a subject or to quickly change the color of some object. This setting is usually best applied locally, using [masks](#) .

#### global vibrance

This affects the chroma dimension of color over the entire image, prioritizing those colors with low chroma. This allows the chroma of neutral colors to be increased without exaggerating already-colorful pixels.

#### contrast

This setting is applied on the luminance channel at constant hue and chroma. The fulcrum setting (in the [masks](#) tab, under *contrast gray fulcrum*) allows you to set the neutral point of the contrast curve:

- at the fulcrum, the contrast curve leaves the luminance unchanged,
- below the fulcrum, the contrast curve decreases the luminance for positive contrast values, or increases it for negative values,
- above the fulcrum, the contrast curve increases the luminance for positive contrast values, or decreases it for negative values.

The fulcrum has a value of 18.45% by default, which is consistent with the current scene-referred workflow and should fit most use cases (assuming that global brightness has been fixed as recommended using the [exposure](#) module).

The contrast algorithm gives natural results that mimic the central part of the contrast curve of analog film. However, it will also increase the image's dynamic range, which may void *filmic* settings in the pipe. For global contrast adjustments, you should normally use the [tone equalizer](#) module - the *color balance RGB* contrast slider is best used with masks, e.g. for selective corrections over the foreground or background.

## linear chroma grading

Linear chroma grading affects the chroma dimension proportionally to its input value, at constant hue and luminance. It does this globally, with a flat coefficient (using the *global chroma*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the [masks](#) tab under *luminance ranges*).



## perceptual saturation grading

Perceptual saturation grading affects both the luminance and the chroma dimensions, in a perceptual space, proportionally to its input value, at constant hue. It does this globally, with a flat coefficient (using the *global saturation*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the [masks](#) tab under *luminance ranges*).

## perceptual brilliance grading

Perceptual brilliance grading affects both the luminance and the chroma dimensions, in a perceptual space, proportionally to its input value, at constant hue, and in a direction orthogonal to the saturation. Its effect is close to that of changing exposure, but scaled perceptually. It does this globally, with a flat coefficient (using the *global saturation*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the [masks](#) tab under *luminance ranges*).

## 4 ways tab

Each of the settings in the 4 ways tab is composed of the same three components, which define a color using independent coordinates:

1. *luminance*,
2. *hue*,
3. *chroma*.

Color input like this defines a color shift applied to the image globally or over the specified luminance range.

Each hue slider has a color-picker, which may be used to compute the opponent color of the selected region. This is useful to revert unwanted color casts (e.g. skin redness), since shifting the color to its opponent cast neutralizes it.

## global offset

This is equivalent to the ASC CDL *offset* and falls back to adding a constant RGB value to all pixels, quite like the *black offset* in the *exposure* module. This control does not use masking.

## shadows lift

This is conceptually equivalent to the *lift* from *lift/gamma/gain*, although implemented differently, and falls back to multiplying the masked pixels by a constant RGB value. It is applied using the *shadows* mask.

## highlights gain

This is equivalent to the ASC CDL *slope*, and falls back to multiplying a the masked pixels by a constant RGB value. It is applied using the *highlights* mask.

## global power

This is equivalent to the ASC CDL *power*, and falls back to applying a constant RGB exponent. It is not masked and needs to be normalized, since the power function has a different behaviour above and below 1, and we are in an unbounded pipeline where white is typically greater than 1. The normalization parameter is available in the [masks](#) tab under *white fulcrum*.

## masks tab

This tab defines auxiliary controls for the previous tabs. Masking controls typically don't require any user modification since the defaults are calibrated to suit most needs and fulfil the normal scene-referred pixel pipeline expectations. You should only need to change these settings in specific scenarios.

## luminance ranges

The graphs show the opacity (on the y axis) of the 3 luminance masks relative to the pixel luminance (on the x axis). The darkest curve represents the *shadows* mask, the brightest represents the *highlights* mask, and the third curve represents the *mid-tones* mask.

Only the *shadows* and *highlights* masks can be controlled directly – the *mid-tones* mask is computed indirectly from the others and acts as an adjustment variable.

### shadows fall-off

Control the softness or hardness of the transition from fully opaque (100%) to fully transparent (0%) for the shadows mask.

### mask middle-gray fulcrum

Set the luminance value where all three masks have 50% opacity. In practice, this is used to define how the image is separated into shadows and highlights.

### highlights fall-off

Control the softness or hardness of the transition from fully opaque (100%) to fully transparent (0%) for the highlights mask.

For each of these settings, a mask button, provided to the right of the slider, displays the appropriate mask (shadows, mid-tones, highlights), overlaid as a checker board. The still-visible area of the image (not hidden by the mask) is the area that will be affected by the shadows, mid-tones and highlights sliders in the other tabs.

All mask previews display the output of the module, including any color changes made, so you can also activate them while editing, to see only the affected part of the image.

Luminance masks are computed at the input of the module, which means that they are insensitive to any luminance changes made inside the module.

## thresholds

### white fulcrum

Set the white point luminance in EV. This is used to normalize the *power* setting in the [4 ways](#) tab. Display-referred implementations of power functions assume that white is at 100%, which removes the need for normalization. For scene-referred purposes this needs to be taken into account.

The color-picker to the right of the slider automatically sets the white fulcrum to the maximum luminance from the selected region, which should be sufficient in most cases.

### contrast gray fulcrum

Set the fulcrum for the *contrast* setting in the [master](#) tab. This corresponds to the luminance value that will be left unchanged by the contrast adjustment. This setting usually matches the middle-gray linear value. If you followed the scene-referred workflow recommendations and set the global brightness early in the pipeline, using the *exposure* module, the correct value should usually be around 18-20%.

The color-picker to the right of the slider automatically sets the contrast gray fulcrum to the average luminance from the selected region. This relies on the assumption that the average luminance is usually close to middle-gray, which is not true if you have specular highlights or primary light sources in the frame, or for low/high-key images.

## mask preview settings

These settings apply to the mask previews, displayed by clicking the mask buttons in the *luminance ranges* section. These settings are saved globally, so will be applied to all subsequent images unless changed.

### checker board color 1 and 2

Set the two colors for the background checker board mask underlay. You can set them to opponent colors of the current image to aid legibility.

### checker board size

Set the width of the checker board cells in pixels (adjusted according to the display's DPI setting).

## FAQ

### saturation or chroma?

As described in the [dimensions of color](#) section, saturation and chroma roam the (lightness, chroma) plane in different directions. In addition, the chroma of *color balance RGB* uses a scene-referred linear space, while the saturation uses a perceptual space, which rescales color for even spacing.

In practice, you should use the chroma setting if you want to preserve the scene-linearity of the light emission and/or keep the luminance unchanged. However, these changes might affect some hues more heavily than others, due to the fact that the color space is not fully perceptually-scaled.

Saturation is closer to the effect of mixing white paint with some base color. Reducing the saturation of red will degrade it to pink, while reducing its chroma will degrade to a gray shade at the same luminance. Saturation is perhaps a more intuitive way to interact with color, due to its connection with painting.

Choosing one or the other is mostly a matter of deciding where on the (lightness, chroma) graph you want to push your colors, and where they are to begin with. To reach pastel colors, saturation is the way to go. To reach laser-like colors (almost monochromatic), at the risk of looking synthetic, chroma is the way to go.

### what is the connection with lift/gamma/gain?

The *lift/gamma/gain* algorithm relies on a display-referred color space, since it assumes a bounded and symmetric dynamic range, with white point at 100% and gray at 50%. As such, it is simply unusable in a scene-referred space. However, the only incompatible part is the *lift*. The *gamma* is exactly the ASC CDL *power*, and the *gain* is exactly the ASC CDL *slope*.

The *color balance RGB* module simply has two slopes instead of one: the *gain*, applied on the highlights extracted from the whole image by a mask, and the *lift*, applied similarly but on the shadows.

### changing contrast

While *color balance RGB* is mostly about color (other modules handle the global contrast in chromaticity-preserving ways) luminance is as much a part of color as hue or chroma, and it needs to be dealt with here too, because the perception of saturation relies on it. If you wish to turn red into pink, for example, reducing its chroma will turn it gray, so you need to increase its luminance as well.

There are several ways to change the contrast in *color balance RGB*, either locally (with masks) or globally (without):

- in the [master](#) tab, use the *contrast* setting (possibly alongside the *contrast gray fulcrum* in the [masks](#) tab). Be aware that this will raise the white point and therefore increase the dynamic range of the image, which may void filmic settings later in the pipeline.
- in [perceptual saturation grading](#), desaturate highlights and resaturate shadows to produce a luminance contrast boost,
- in [perceptual brilliance grading](#), add brilliance in the highlights and remove brilliance in the shadows to produce to a luminance contrast boost,
- in the [4 ways](#) tab, set the *shadows lift* luminance to negative values and the *highlights gain* luminance to positive values, which also produces a luminance contrast boost.

The difference between these methods is how the effect will be weighted relative to the input of the module. You are advised to do the majority of luminance contrast adjustments in the *filmic* and *tone equalizer* modules, and then undertake final changes in *color balance RGB* while examining the colors.

## internal processing

The following is the internal order of operations within the module:

1. Transform from pipeline RGB to Kirk/FilmLight Ych space,
2. Apply hue shift at constant chroma and constant luminance,
3. Compute luminance masks with Y,
4. Apply the *linear chroma* and *vibrance* settings at constant hue and luminance,
5. Transform to Kirk/FilmLight RGB space,
6. Apply the *4 ways* settings (except *luminance power*),
7. Transform to Kirk/FilmLight Yrg space,
8. Apply *luminance power* and *contrast* on Y,
9. Transform to JzAzBz space,
10. Apply the *perceptual saturation* and *perceptual brilliance* settings,
11. Soft-clip the chroma using pipeline RGB gamut at constant hue and lightness,
12. Transform back to pipeline RGB.

## caveats

Setting the global chroma to -100% will not produce a real monochrome image, as is customary with other algorithms. The reason for this is that the RGB space used has a D65 white point defined in CIE LMS 2006 space, while darktable uses a white point defined in CIE XYZ 1931 space, and there is no exact conversion between these spaces. The result will therefore be a slightly tinted black & white image. If your intent is to get a real black & white image using the luminance channel, the *color calibration* module offers a *B&W : luminance-based* preset that does exactly the same thing but without the white-point discrepancy.

This module has its gamut-mapping (against pipeline RGB) permanently enabled. This means that if your original image contains some largely out-of-gamut colors to start with, simply enabling *color balance RGB* with no particular setting will slightly alter its colors. This is probably for the best.

The maximum saturation allowed in the pipeline working RGB space is recorded for each hue when initializing the module, and is later cached in a LUT (look-up table) to save performance. If the working profile is later changed, *color balance RGB* is not notified, meaning that it will not update its cached hue/saturation LUT. To force a LUT update, you can simply change any setting in the *color balance RGB* module, then change it back again. It is not recommended that you change the working RGB space half-way through an editing session, as this could result in unexpected chroma and hue changes.

For performance reasons, the non-linear conversions from and to the working RGB space are bypassed, meaning that the internal colorimetry will be wrong when using non-linear color spaces. Note that there is no reason to use non-linear spaces as working RGB since they make alpha blending more challenging for no benefit.

- 
1. Richard A. Kirk, Chromaticity coordinates for graphic arts based on CIE 2006 LMS with even spacing of Munsell colours, 2019. <https://doi.org/10.2352/jissn.2169-2629.2019.27.38> ↵
  2. Safdar et al., Perceptually uniform color space for image signals including high dynamic range and wide gamut, 2017. <https://doi.org/10.1364/OE.25.015131> ↵

## 8.2.11. color calibration

A fully-featured color-space correction, white balance adjustment and channel mixer module.

This simple yet powerful module can be used in the following ways:

- To adjust the white balance ( [chromatic adaptation](#) ), working in tandem with the [white balance](#) module. Here, the *white balance* module makes some initial adjustments (required for the [demosaic](#) module to work effectively), and the *color calibration* module then calculates a more perceptually-accurate white balance after the input color profile has been applied.
- As a simple RGB [channel mixer](#) , adjusting the R, G and B output channels based on the R, G and B input channels, to perform cross-talk color-grading.
- To adjust the [color saturation and brightness](#) of the image, based on the relative strength of the R, G and B channels of each pixel.
- To produce a [grayscale output](#) based on the relative strengths of the R, G and B channels, in a way similar to the response of black and white film to a light spectrum.
- To improve the color accuracy of the input color profile using a [color checker](#) chart.

### White Balance in the Chromatic Adaptation Transformation (CAT) tab

Chromatic adaptation aims to predict how all surfaces in the scene would look if they had been lit by another illuminant. What we actually want to predict, though, is how those surfaces would have looked if they had been lit by the same illuminant as your monitor, in order to make all colors in the scene match the change of illuminant. White balance, on the other hand, aims only at ensuring that whites and grays are really neutral ( $R = G = B$ ) and doesn't really care about the rest of the color range. White balance is therefore only a partial chromatic adaptation.

Chromatic adaptation is controlled within the Chromatic Adaptation Transformation (CAT) tab of the *color calibration* module. When used in this way the *white balance* module is still required as it needs to perform a basic white balance operation (connected to the input color profile values). This technical white balancing ("camera reference" mode) is a flat setting that makes grays lit by a standard D65 illuminant look achromatic, and makes the demosaicing process more accurate, but does not perform any perceptual adaptation according to the scene. The actual chromatic adaptation is then performed by the *color calibration* module, on top of those corrections performed by the *white balance* and *input color profile* modules. The use of custom matrices in the *input color profile* module is therefore discouraged. Additionally, the RGB coefficients in the *white balance* module need to be accurate in order for this module to work in a predictable way.

The *color calibration* and *white balance* modules can be automatically applied to perform chromatic adaptation for new edits by setting the chromatic adaptation workflow option ( [preferences > processing > auto-apply chromatic adaptation defaults](#) ) to "modern". If you prefer to perform all white balancing within the *white balance* module, a "legacy" option is also provided. Neither option precludes the use of other modules (such as [color balance RGB](#) ) for creative color grading further along the pixel pipeline.

By default, *color calibration* performs chromatic adaptation by:

- reading the RAW file's Exif data to fetch the scene white balance set by the camera,
- adjusting this setting using the camera reference white balance from the *white balance* module,
- further adjusting this setting with the input color profile in use (standard matrix only).

For consistency, the *color calibration* module's default settings always assume that the standard matrix is used in the *input color profile* module - any non-standard settings in this module are ignored. However, *color calibration*'s defaults can read any auto-applied preset in the *white balance* module.

It is also worth noting that, unlike the *white balance* module, *color calibration* can be used with [masks](#) . This means that you can selectively correct different parts of the image to account for differing light sources.

To achieve this, create an instance of the *color calibration* module to perform global adjustments using a mask to exclude those parts of the image that you wish to handle differently. Then create a second instance of the module reusing the mask from the first instance (inverted) using a [raster mask](#) .

## CAT tab workflow

The default illuminant and color space used by the chromatic adaptation are initialised from the Exif metadata of the RAW file. There are four options available in the CAT tab to set these parameters manually:

- Use the color-picker (to the right of the color patch) to select a neutral color from the image or, if one is unavailable, select the entire image. In this case, the algorithm finds the average color within the chosen area and sets that color as the illuminant. This method relies on the “gray-world” assumption, which predicts that the average color of a natural scene will be neutral. This method will not work for artificial scenes, for example those with painted surfaces.
- Select “(AI) detect from edges”, which uses a machine-learning technique to detect the illuminant using the entire image. This algorithm finds the average gradient color over the edges in the image and sets that color as the illuminant. This method relies on the “gray-edge” assumption, which may fail if large chromatic aberrations are present. As with any edge-detection method, it is sensitive to noise and poorly suited to high-ISO images, but it is very well suited for artificial scenes where no neutral colors are available.
- Select “(AI) detect from surfaces”, which combines the two previous methods, also using the entire image. This algorithm finds the average color within the image, giving greater weight to areas where sharp details are found and colors are strongly correlated. This makes it more immune to noise than the *edge* variant and more immune to legitimate non-neutral surfaces than the naïve average, but sharp colored textures (like green grass) are likely to make it fail.
- Select “as shot in camera” to restore the camera defaults and re-read the RAW Exif.

The color patch shows the color of the currently calculated illuminant projected into sRGB space. The aim of the chromatic adaptation algorithm is to turn this color into pure white, which does not necessarily mean shifting the image toward its *perceptual* opponent color. If the illuminant is properly set, the image will be given the same tint as shown in the color patch when the module is disabled.

To the left of the color patch is the CCT (correlated color temperature) approximation. This is the closest temperature, in kelvin, to the illuminant currently in use. In most image processing software it is customary to set the white balance using a combination of temperature and tint. However, when the illuminant is far from daylight, the CCT becomes inaccurate and irrelevant, and the CIE (International Commission on Illumination) discourages its use in such conditions. The CCT reading informs you of the closest CCT match found:

- When the CCT is followed by “(daylight)”, this means that the current illuminant is close to an ideal daylight spectrum  $\pm 0.5\%$ , and the CCT figure is therefore meaningful. In this case, you are advised to use the “D (daylight)” illuminant.
- When the CCT is followed by “(black body)”, this means that the current illuminant is close to an ideal black body (Planckian) spectrum  $\pm 0.5\%$ , and the CCT figure is therefore meaningful. In this case, you are advised to use the “Planckian (black body)” illuminant.
- When the CCT is followed by “(invalid)”, this means that the CCT figure is meaningless and wrong, because we are too far from either a daylight or a black body light spectrum. In this case, you are advised to use the *custom* illuminant. The chromatic adaptation will still perform as expected (see the note below), so the “(invalid)” tag only means that the current illuminant color is not accurately tied to the displayed CCT. This tag is nothing to be concerned about – it is merely there to tell you to stay away from the daylight and planckian illuminants because they will not behave as you might expect.

When one of the above illuminant detection methods is used, the module checks where the calculated illuminant sits using the two idealized spectra (daylight and black body) and chooses the most accurate spectrum model to use in its *illuminant* parameter. The user-interface will change accordingly:

- A temperature slider will be provided if the detected illuminant is close to a D (daylight) or Planckian (black body) spectrum, for which the CCT is meaningful.
- Hue and chroma sliders in CIE 1976 Luv space are offered for the *custom* illuminant, which allows direct selection of the illuminant color in a perceptual framework without any intermediate assumption.

---

**Note:** Internally, the illuminant is represented by its absolute chromaticity coordinates in CIE xyY color space. The illuminant selection options in the module are merely interfaces to set up this chromaticity from real-world relationships and are intended to make this process faster. It does not matter to the algorithm if the CCT is tagged “invalid” – this just means that the relationship between the CCT and the corresponding xyY coordinates is not physically accurate. Regardless, the color set for the illuminant, as displayed in the patch, will always be honored by the algorithm.

When switching from one illuminant to another, the module attempts to translate the previous settings to the new illuminant as accurately as possible. Switching from any illuminant to *custom* preserves your settings entirely, since the *custom* illuminant is a general case. Switching between other modes, or from *custom* to any other mode, will not precisely preserve your settings from the previous mode due to rounding errors.

Other hard-coded *illuminants* are available (see below). Their values come from standard CIE illuminants and are absolute. You can use them directly if you know exactly what kind of light bulb was used to illuminate the scene and if you trust your camera's input profile and reference (D65) coefficients to be accurate. Otherwise, see [caveats](#) below.

## CAT tab controls

### adaptation

The working color space in which the module will perform its chromatic adaptation transform and channel mixing. The following options are provided:

- *Linear Bradford (1985)*: This is accurate for illuminants close to daylight and is compatible with the ICC v4 standard, but produces out-of-gamut colors for more difficult illuminants.
- *CAT16 (2016)*: This is the default option and is more robust in avoiding imaginary colors while working with large gamut or saturated cyan and purple. It is more accurate than the Bradford CAT in most cases.
- *Non-linear Bradford (1985)*: This can sometimes produce better results than the linear version but is unreliable.
- *XYZ*: This is the least accurate method and is generally not recommended except for testing and debugging purposes.
- *none (disable)*: Disable any adaptation and use the pipeline working RGB space.

### illuminant

The type of illuminant assumed to have lit the scene. Choose from the following:

- *same as pipeline (D50)*: Do not perform chromatic adaptation in this module instance but just perform channel mixing, using the selected *adaptation* color space.
- *CIE standard illuminant*: Choose from one of the CIE standard illuminants (daylight, incandescent, fluorescent, equi-energy, or black body), or a non-standard "LED light" illuminant. These values are all pre-computed – as long as your camera sensor is properly profiled, you can just use them as-is. For illuminants that lie near the Planckian locus, an additional "temperature" control is also provided (see below).
- *custom*: If a neutral gray patch is available in the image, the color of the illuminant can be selected using the color picker, or can be manually specified using hue and saturation sliders (in LCh perceptual color space). The color swatch next to the color picker shows the color of the calculated illuminant used in the CAT compensation. The color picker can also be used to restrict the area used for AI detection (below).
- *(AI) detect from image surfaces*: This algorithm obtains the average color of image patches that have a high covariance between chroma channels in YUV space and a high intra-channel variance. In other words, it looks for parts of the image that appear as though they should be gray, and discards flat colored surfaces that may be legitimately non-gray. It also discards chroma noise as well as chromatic aberrations.
- *(AI) detect from image edges*: Unlike the *white balance* module's auto-white-balancing which relies on the "gray world" assumption, this method auto-detects a suitable illuminant using the "gray edge" assumption, by calculating the Minkowski p-norm ( $p = 8$ ) of the laplacian and trying to minimize it. That is to say, it assumes that edges should have the same gradient over all channels (gray edges). It is more sensitive to noise than the previous surface-based detection method.
- *as shot in camera*: Calculate the illuminant based on the white balance settings provided by the camera.

### temperature

Adjust the color temperature of the illuminant. Move the slider to the right to assume a more blue illuminant, which will make the white-balanced image appear warmer/more red. Move the slider to the left to assume a more red illuminant, which makes the image appear cooler/more blue after compensation.

This control is only provided for illuminants that lie near the Planckian locus and provides fine-adjustment along that locus. For other illuminants the concept of "color temperature" doesn't make sense, so no temperature slider is provided.

### hue

For custom white balance, set the *hue* of the illuminant color in LCh color space (derived from CIE Luv space).

### chroma

For custom white balance, set the *chroma* (or saturation) of the illuminant color in LCh color space (derived from CIE Luv space).



### gamut compression

Most camera sensors are slightly sensitive to invisible UV wavelengths, which are recorded on the blue channel and produce “imaginary” colors. Once corrected by the input color profile, these colors will end up out of gamut (that is, it may no longer be possible to represent certain colors as a valid [R,G,B] triplet with positive values in the working color space) and produce visual artifacts in gradients. The chromatic adaptation may also push other valid colors out of gamut, at the same time pushing any already out-of-gamut colors even further out of gamut.

*Gamut compression* uses a perceptual, non-destructive, method to attempt to compress the chroma while preserving the luminance as-is and the hue as close as possible, in order to fit the whole image into the gamut of the pipeline working color space. One example where this feature is very useful is for scenes containing blue LED lights, which are often quite problematic and can result in ugly gamut clipping in the final image.

### clip negative RGB from gamut

Remove any negative RGB values (set them to zero). This helps to deal with bad black level as well as the blue channel clipping issues that may occur with blue LED lights. This option is destructive for color (it may change the hue) but ensures a valid RGB output no matter what. It should never be disabled unless you want to take care of the gamut mapping manually and understand what you are doing. In that case, use the *black level correction* in the *exposure* module to get rid of any negative RGB (RGB means light, which is energy, and which should always be a positive quantity), then increase the *gamut compression* until no solid black patches remain in the image. Proper denoising may help getting rid of odd RGB values too. Note that this approach may still be insufficient to recover some deep and luminous shades of blue.

---

**Note 1:** It has been reported that some OpenCL drivers don't play well when negative RGB are present in the pixel pipeline, because many pixel operators use logarithms and power functions (*filmic*, *color balance*, all the CIE Lab <-> CIE XYZ color space conversions), which are not defined for negative numbers. Although the inputs are sanitized before sensitive operations, it is not enough for some OpenCL drivers, which will output isolated NaN (Not a Number) values. These NaN values may be subsequently spread by local filters (blurring and sharpening operations, like *sharpness*, *local contrast*, *contrast equalizer*, *low pass*, *high pass*, *surface blur*, and *filmic* highlights reconstruction), resulting in large black, grey or white squares.

In all these cases, you **must** enable the “clip negative RGB from gamut” option in the *color calibration* module.

**Note 2:** A common case for failure of the color algorithms in *color calibration* (especially the gamut compression) is pixels that have a luminance value of 0 (Y channel of the CIE 1931 XYZ space), but non-zero chromaticity values (X and Z channels of the CIE 1931 XYZ space). This case is a numerical oddity that matches no physical reality (a pixel with no luminance should have no chromaticity either), will produce a division by zero in xyY and Yuv color spaces, and will create NaN RGB values as a result. This issue is **not** corrected inside *color calibration* because it is a symptom of a bad input profiling and/or a bad black point level, and needs to be addressed manually either by adjusting the input color profile with the channel mixer or in the *exposure* module's *black level correction*.

---



## CAT warnings

The chromatic adaptation in this module relies on a number of assumptions about the earlier processing steps in the pipeline in order to work correctly, and it can be easy to inadvertently break those assumptions in subtle ways. To help you to avoid these kinds of mistakes, the *color calibration* module will show warnings in the following circumstances.

- If the *color calibration* module is set up to perform chromatic adaptation but the *white balance* module is not set to “camera reference”, warnings will be shown in both modules. These errors can be resolved either by setting the *white balance* module to “camera reference” or by disabling chromatic adaptation in the *color calibration* module. Note that some sensors may require minor corrections within the *white balance* module in which case these warnings can be ignored.
- If two or more instances of *color calibration* have been created, each attempting to perform chromatic adaptation, an error will be shown on the second instance. This could be a valid use case (for instance where masks have been set up to apply different white balances to different non-overlapping areas of the image) in which case the warnings can be ignored. For most other cases, chromatic adaptation should be disabled in one of the instances to avoid double-corrections.

By default, if an instance of the *color calibration* module is already performing chromatic adaptation, each new instance you create will automatically have its adaptation set to “none (bypass)” to avoid this “double-correction” error.

The chromatic adaptation modes in *color calibration* can be disabled by either setting the *adaptation* to “none (bypass)” or setting the *illuminant* to “same as pipeline (D50)” in the CAT tab.

These warnings are intended to prevent common and easy mistakes while using the automatic default presets in the module in a typical RAW editing workflow. When using custom presets and some specific workflows, such as editing film scans or JPEGs, these warnings can and should be ignored.

Advanced users can disable module warnings in [preferences > processing > show warning messages](#) .

## channel mixing

The remainder of this module is a standard channel mixer, allowing you to adjust the output R, G, B, colorfulness, brightness and gray of the module based on the relative strengths of the R, G and B input channels.

Channel mixing is performed in the color space defined by the *adaptation* control on the [CAT tab](#) . For all practical purposes, these CAT spaces are particular RGB spaces tied to human physiology and proportional to the light emissions in the scene, but they still behave in the same way as any other RGB space. The use of any of the CAT spaces can make the channel mixer tuning process easier, due to their connection with human physiology, but it is also possible to mix channels in the RGB working space of the pipeline by setting the *adaptation* to “none (bypass)”. To perform channel mixing in one of the *adaptation* color spaces without chromatic adaptation, set the *illuminant* to “same as pipeline (D50)”.

---

**Note:** The actual colors of the CAT or RGB primaries used for the channel mixing, projected to sRGB display space, are painted in the background of the RGB sliders, so you can get a sense of the color shift that will result from your altered settings.

Channel mixing is a process that defines a boosting/muting factor for each channel as a ratio of all the original channels. Instead of entering a single flat correction that ties the output value of a channel to its input value (for example,  $R_{\text{output}} = R_{\text{input}} \times \text{correction}$ ), the correction to each channel is dependent on the input of *all* of the channels for each pixel (for example,  $R_{\text{output}} = R_{\text{input}} \times R_{\text{correction}} + G_{\text{input}} \times G_{\text{correction}} + B_{\text{input}} \times B_{\text{correction}}$ ). Thus a pixel's channels contribute to each other (a process known as "cross-talk") which is equivalent to rotating the primary colors of the color space in 3D. This is, in effect, digital simulation of physical color filters.

Although rotating primary colors in 3D is ultimately equivalent to applying a general hue rotation, the connection between the RGB corrections and the resulting perceptual hue rotation is not directly predictable, which makes the process non-intuitive. "R", "G" and "B" should be taken as a mixture of 3 lights that we dial up and down, not as a set of colors or hues. Also, since RGB tristimulus does not decouple luminance and chrominance, but is an additive lighting setup, the "G" channel is more strongly tied to human luminance perception than the "R" and "B" ones. All pixels have a non-zero G channel, which implies that any correction to the G channel is likely to affect all pixels.

The channel mixing process is therefore tied to a physical interpretation of the RGB tristimulus (as additive lights), which makes it well-suited for primary color grading and illuminant corrections, and blends the color changes smoothly. However, trying to understand and predict it from a perceptual point of view (luminance, hue and saturation) is going to fail and is discouraged.

**Note:** The "R", "G" and "B" labels on the channels of the color spaces in this module are merely conventions formed out of habit. These channels do not necessarily look "red", "green" and "blue", and users are advised against trying to make sense out of them based on their names. This is a general principle that applies to *any* RGB space used in any application.

## R, G and B tabs

At its most basic level, you can think of the R, G and B tabs of the *color calibration* module as a type of matrix multiplication between a 3x3 matrix and the input [R G B] values. This is in fact very similar to what a matrix-based ICC color profile does, except that the user can input the matrix coefficients via the darktable GUI rather than reading the coefficients from an ICC profile file.

$$\begin{bmatrix} R_{\text{out}} \\ G_{\text{out}} \\ B_{\text{out}} \end{bmatrix} = \begin{bmatrix} R_r & R_g & R_b \\ G_r & G_g & G_b \\ B_r & B_g & B_b \end{bmatrix} \times \begin{bmatrix} R_{\text{in}} \\ G_{\text{in}} \\ B_{\text{in}} \end{bmatrix}$$

If, for example, you've been provided with a matrix to transform from one color space to another, you can enter the matrix coefficients into the *channel mixer* as follows:

- select the *R* tab and then set the  $R_r$ ,  $R_g$  &  $R_b$  values using the R, G and B input sliders
- select the *G* tab and then set the  $G_r$ ,  $G_g$  &  $G_b$  values using the R, G and B input sliders
- select the *B* tab and then set the  $B_r$ ,  $B_g$  &  $B_b$  values using the R, G and B input sliders

By default, the mixing function in *color calibration* just copies the input [R G B] channels straight over to the matching output channels. This is equivalent to multiplying by the identity matrix:

$$\begin{bmatrix} R_{\text{out}} \\ G_{\text{out}} \\ B_{\text{out}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} R_{\text{in}} \\ G_{\text{in}} \\ B_{\text{in}} \end{bmatrix}$$

For a more intuitive understanding of how the mixing sliders on the R, G and B tabs behave, consider the following:

- for the *R* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "red". Moving the slider to the left will make those areas more "cyan".
- for the *G* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "green". Moving the slider to the left will make those areas more "magenta".
- for the *B* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "blue". Moving the slider to the left will make those areas more "yellow".

## R, G and B tab controls

The following controls are shown for each of the R, G and B tabs:

### input R/G/B

Choose how much the input R, G and B channels influence the output channel relating to the tab concerned.

**normalize channels**

Select this checkbox to normalize the coefficients to try to preserve the overall brightness of this channel in the final image as compared to the input image.

## brightness and colorfulness tabs

The brightness and colorfulness (color saturation) of pixels in an image can also be adjusted based on the R, G and B input channels. This uses the same basic algorithm that the [filmic rgb](#) module uses for tone mapping (which preserves RGB ratios) and for mid-tones saturation (which massages them).

**saturation algorithm**

This control allows you to upgrade the saturation algorithm to the new 2021 version, for edits produced prior to darktable 3.6 - it will not appear for edits that already use the latest version.

## colorfulness tab controls

**input R/G/B**

Adjust the color saturation of pixels, based on the R, G and B channels of those pixels. For example, adjusting the *input R* slider will affect the color saturation of pixels containing a lot of "R" more than pixels containing only a small amount of "R".

**normalize channels**

Select this checkbox to try to keep the overall saturation constant between the input and output images.

## brightness tab controls

**input R/G/B**

Adjust the brightness of certain colors in the image, based on the R, G and B channels of those colors. For example, adjusting the *input R* slider will affect the brightness of colors containing a lot of R channel much more than colors containing only a small amount of R channel. When darkening/brightening a pixel, the ratio of the R, G and B channels for that pixel is maintained, in order to preserve the hue.

**normalize channels**

Select this checkbox to try to keep the overall brightness constant between the input and output images.

## gray tab

Another very useful application of *color calibration* is the ability to mix the channels together to produce a grayscale output - a monochrome image. Select the *gray* tab, and set the R, G and B sliders to control how much each channel contributes to the brightness of the output. This is equivalent to the following matrix multiplication:

$$\text{GRAY\_out} = [r \ g \ b] \times \begin{bmatrix} R\_in \\ G\_in \\ B\_in \end{bmatrix}$$

When dealing with skin tones, the relative weights of the three channels will affect the level of detail in the image. Placing more weight on R (e.g. [0.9, 0.3, -0.3]) will make for smooth skin tones, whereas emphasising G (e.g. [0.4, 0.75, -0.15]) will bring out more detail. In both cases the B channel is reduced to avoid emphasising unwanted skin texture.

## gray tab controls

**input R/G/B**

Choose how much each of the R, G and B channels contribute to the gray level of the output. The image will only be converted to monochrome if the three sliders add up to some non-zero value. Adding more B will tend to bring out more details, adding more R will tend to smooth skin tones.

**normalize channels**

Select this checkbox to try to keep the overall brightness constant as the sliders are adjusted.

## extracting settings using a color checker

Since the channel mixer is essentially an RGB matrix (similar to the [input color profile](#) used for RAW images) it can be used to improve the color accuracy of the input color profile by computing ad-hoc color calibration settings.

These computed settings aim to minimize the color difference between the scene reference and the camera recording in a given lighting situation. This is equivalent to creating a generic ICC color profile but here, the profile is instead stored as module settings that can be saved as presets or styles, to be shared and re-used between images. Such profiles are meant to complement and refine the generic input profile but do not replace it.

This feature can assist with:

- handling difficult illuminants, such as low [CRI](#) light bulbs, for which a mere white balancing will never suffice,
- digitizing artworks or commercial products where an accurate rendition of the original colors is required,
- neutralizing a number of different cameras to the same ground-truth, in multi-camera photo sessions, in order to obtain a consistent base look and share the color editing settings with a consistent final look,
- obtaining a sane color pipeline from the start, nailing white balance and removing any bounced-light color cast at once, with minimal effort and time.

## supported color checker targets

Users are not currently permitted to use custom targets, but a limited number of verified color checkers (from reputable manufacturers) are supported:

- X-Rite / Gretag MacBeth Color Checker 24 (pre- and post-2014),
- Datacolor SpyderCheckr 24,
- Datacolor SpyderCheckr 48.

Users are discouraged from obtaining cheap, off-brand, color targets as color constancy between batches cannot possibly be asserted at such prices. Inaccurate color checkers will only defeat the purpose of color calibration and possibly make things worse.

IT7 and IT8 charts are not supported since they are hardly portable and not practical for use on-location for ad-hoc profiles. These charts are better suited for creating generic color profiles, undertaken using a standard illuminant, for example with [Argyll CMS](#) .

---

**Note:** X-Rite changed the formula of their pigments in 2014, which slightly altered the color of the patches. Both formulas are supported in darktable, but you should be careful to choose the correct reference for your target. If in doubt, try both and choose the one that yields the lowest average delta E after calibration.

---

## prerequisites

In order to use this feature you will need to take a test shot of a supported color checker chart, on-location, under appropriate lighting conditions:

- frame the chart in the center 50% of the camera's field, to ensure that the image is free of vignetting,
- ensure that the main light source is far enough from the chart to give an even lighting field over the surface of the chart,
- adjust the angle between the light, chart and lens to prevent reflections and gloss on the color patches,
- for the best quality profile you should capture an image with the appropriate brightness. To achieve this, take a few bracketed images (between -1 and +1 EV) of your color checker and load them into darktable, ensuring that all modules between *color calibration* and *output color profile* are disabled. Choose the image where the white patch has a brightness L of 94-96% in CIE Lab space or a luminance Y of 83-88% in CIE XYZ space (use the global color picker). This step is not strictly necessary - alternatively you can take a single image and apply the exposure compensation as recommended in the profile report.

If the lighting conditions are close to a standard D50 to D65 illuminant (direct natural light, no colored bounced light), the color checker shot can be used to produce a generic profile that will be suitable for any daylight illuminant with only a slight adjustment of the white balance.

If the lighting conditions are peculiar and far from standard illuminants, the color checker shot will be only usable as an ad-hoc profile for pictures taken in the same lighting conditions.

## usage

The settings used in color calibration depend on the chosen CAT space and on any color settings defined earlier in the pipe within the *white balance* and *input color profile* modules. As such, the results of the profiling (e.g. the RGB channel mixing coefficients) are valid only for a rigid set of *CAT space*, *white balance* and *input color profile* settings. If you wish to create a generic style with your profile, don't forget that you will need to include the settings from these modules as well.

Use the following process to create your profile preset/style:

1. Enable the [lens correction](#) module to correct any vignetting that might mislead the calibration,
2. On the bottom of the *color calibration* module, click on the arrow near the *calibrate with a color checker* label, to show the controls,
3. Pick the correct model and manufacturer of your color checker from the *chart* drop-down,
4. In the image preview, an overlay of the chart's patches will appear. Drag the corners of the chart so that they match the visual references (dots or crosses) around the target, to compensate for any perspective distortion,
5. Click the *refresh* button to compute the profile,
6. Check the *Profile quality report*. If it is "good", you can click on the *validation* button. If not, try changing the optimization strategy and refresh the profile again.
7. Save the profile in a preset or style, or simply copy & paste the module settings to all of the pictures taken under the same lighting conditions, from within the lighttable view or filmstrip.

---

**Note:** You don't *need* to use the standard matrix in the *input color profile* module when performing a calibration, but be aware that the "as shot in camera" default white balance will not work properly with any other profile, and that you will need to always use the same input profile whenever you reuse such calibration settings.

---

## reading the profile report

The profile report helps you to assess the quality of the calibration. The settings in color calibration are only a "best fit" optimization and will never be 100% accurate for the whole color spectrum. We therefore need to track "how inaccurate" it is in order to know whether we can trust this profile or not.

Bad profiles can happen and will do more harm than good if used.

## delta E and the quality report

The [CIE delta E 2000](#) ( $\Delta E$ ) is used as a perceptual metric of the error between the reference color of the patches and the color obtained after each step of calibration:

- $\Delta E = 0$  means that there is no error - the obtained color is exactly the reference color. Unfortunately, this will never happen in practice.
- $\Delta E = 2.3$  is defined as the Just Noticeable Difference (JND).
- $\Delta E < 2.3$  means that the average observer will not be able to tell the difference between the expected reference color and the obtained color. This is a satisfactory result.
- $\Delta E > 2.3$  means that the color difference between the expected reference and the obtained color is noticeable for the average observer. This is unsatisfactory but sometimes unavoidable.

The quality report tracks the average and maximum  $\Delta E$  at the input of the module (before anything is done), after the chromatic adaptation step (white balance only), and at the output of the module (white balance and channel mixing). At each step, the  $\Delta E$  should be lower than at the previous step, if everything goes as planned.

## profile data

The data generated by the profiling process comprises the RGB 3×3 matrix and the detected illuminant. These are expressed in the CAT *adaptation* space defined in the *CAT* tab and are provided in case you want to export these coefficients to other software. If the detected illuminant is *daylight* or *black body*, the matrix should be fairly generic and re-usable for other *daylight* and *black body* illuminants with the addition of a small white balance adjustment.

## normalization values

These are the settings that you should define, as-is, for the *exposure* and *black level correction* parameters in the [exposure](#) module, in order to obtain the lowest possible error in your profile. This step is optional and is useful only when the utmost precision is required, but beware that it can produce negative RGB values that will be clipped at various points in the pipeline.

## overlay



The chart overlay displays a disc in the center of each color patch, which represents the expected reference value of that patch, projected into the display RGB space. This helps you to visually assess the difference between the reference and the actual color without having to bother with  $\Delta E$  values. This visual clue will be reliable only if you set the *exposure* module as instructed in the *normalization values* of the profile report.

Once the profile has been calibrated, some of the square patches will be crossed in the background by one or two diagonals:

- patches that are not crossed have  $\Delta E < 2.3$  (JND), meaning they are accurate enough that the average observer will be unable to notice the deviation,
- patches crossed with one diagonal have  $2.3 < \Delta E < 4.6$ , meaning that they are mildly inaccurate,
- patches crossed with two diagonals have  $\Delta E > 4.6$  ( $2 \times$  JND), meaning that they are highly inaccurate.

This visual feedback will help you to set up the optimization trade-off to check which colors are more or less accurate.



## enhancing the profile

Because any calibration is merely a “best fit” optimization (using a weighted least-squares method) it is impossible to have all patches within our  $\Delta E < 2.3$  tolerance. Some compromise will therefore be required.

The *optimize for* parameter allows you to define an optimization strategy that attempts to increase the profile accuracy in some colors at the expense of others. The following options are available:

- *none*: Don't use an explicit strategy but rely on the implicit strategy defined by the color checker manufacturer. For example, if the color checker has mostly low-saturation patches, the profile will be more accurate for less-saturated colors.
- *neutral colors*: Give priority to grays and less-saturated colors. This is useful for desperate cases involving cheap fluorescent and LED lightings, having low CRI. However, it may increase the error in highly-saturated colors more than not having any profile.
- *saturated colors*: Give priority to primary colors and highly-saturated colors. This is useful in product and commercial photography, to get brand colors right.
- *skin and soil colors, foliage colors, sky and water colors*: Give priority to the chosen hue range. This is useful if the subject of your pictures is clearly defined and has a typical color.
- *average delta E*: Attempt to make the color error uniform across the color range and minimize the average perceptual error. This is useful for generic profiles.
- *maximum delta E*: Attempt to minimize outliers and large errors, at the expense of the average error. This can be useful to get highly saturated blues back into line.

No matter what you do, strategies that favor a low average  $\Delta E$  will usually have a higher maximum  $\Delta E$ , and vice versa. Also, blues are always the more challenging color range to get correct, so the calibration usually falls back to protecting blues at the expense of everything else, or everything else at the expense of blues.

The ease of obtaining a proper calibration depends on the quality of the scene illuminant (daylight and high CRI illuminants should always be preferred), the quality of the primary *input color profile*, the *black point compensation* set in the *exposure* module, but first and foremost on the mathematical properties of the camera sensor's filter array.

## profile checking

It is possible to use the *color space check* button (first on the left, at the bottom of the module) to perform a single  $\Delta E$  computation of the color checker reference against the output of the *color calibration* module. This can be used in the following ways:

1. To check the accuracy of a profile calculated in particular conditions against a color checker shot in different conditions.
2. To evaluate the performance of any color correction performed earlier in the pipe, by setting the *color calibration* parameters to values that effectively disable it (*CAT adaptation* to *none*, everything else set to default), and just use the average  $\Delta E$  as a performance metric.

## caveats

The ability to use standard CIE illuminants and CCT-based interfaces to define the illuminant color depends on sound default values for the standard matrix in the *input color profile* module as well as reasonable RGB coefficients in the *white balance* module.

Some cameras, most notably those from Olympus and Sony, have unexpected white balance coefficients that will always make the detected CCT invalid even for legitimate daylight scene illuminants. This error most likely comes from issues with the standard input matrix, which is taken from the Adobe DNG Converter.

It is possible to alleviate this issue, if you have a computer screen calibrated for a D65 illuminant, using the following process:

1. Display a white surface on your screen, for example by opening a blank canvas in any photo editing software you like
2. Take a blurry (out of focus) picture of that surface with your camera, ensuring that you don't have any “parasite” light in the frame, you have no clipping, and are using an aperture between f/5.6 and f/8,
3. Open the picture in darktable and extract the white balance by using the spot tool in the *white balance* module on the center area of the image (non-central regions might be subject to chromatic aberrations). This will generate a set of 3 RGB coefficients.
4. [Save a preset](#) for the *white balance* module with these coefficients and auto-apply it to any color RAW image created by the same camera.

## 8.2.12. color contrast

A simplified control for changing the contrast or separation of colors between the green/magenta and blue/yellow axes in Lab color space. Higher values increase color contrast, lower values decrease it.

### module controls

#### green-magenta contrast

Change the green-magenta color contrast. This is equivalent to raising or lowering the steepness of the  $a^*$  curve in Lab. Lower values desaturate greens and magenta, while higher values increase their saturation.

#### blue-yellow contrast

Change the blue-yellow color contrast. This is equivalent to raising or lowering the steepness of the  $b^*$  curve in Lab. Lower values desaturate blues and yellows, while higher values increase their saturation.

## 8.2.13. color correction

Modify the global image saturation to give the image a tint or as an alternative method to split-tone the image.

---

**Note:** Use the [color balance rgb](#) module for color modifications.

---

### module controls

#### color board

For split toning move the white dot to the desired highlight tint and then select a tint for shadows with the dark spot. For a simple global tint set both spots to the same color.

#### saturation

Correct the global saturation.

## 8.2.14. color look up table

A generic color look up table implemented in Lab space.

The input to this module is a list of source and target points and the complete mapping is interpolated using splines. The resulting look up tables (LUTs) are editable by hand and can be created using the `darktable-chart` utility to match given input (such as `hald-cluts` and `RAW/JPEG` with in-camera processing pairs). See [using darktable-chart](#) for details.

### module controls

#### color board

The color board grid shows a list of colored patches. The colors of the patches are the source points. The target color of the selected patch is shown as offsets which are controlled by sliders beneath the color board. An outline is drawn around patches that have been altered (where the source and target colors differ).

Click a patch to select it, or use the combo box or color picker. The currently-selected patch is marked with a white square, and its number is displayed in the combo box below.

By default, the module will load the 24 patches of a classic color checker and initialise the mapping to identity (no change to the image). Configurations with more than 24 patches are shown in a 7x7 grid.



### interaction

To modify the color mapping, you can change the source and target colors, though the main use is to change the target colors.

Start with an appropriate palette of source colors (either from the presets menu or from a style you have downloaded). You can then change the lightness (L), green-red (a), blue-yellow (b), or saturation (c) of the patches' target values with the sliders.

To change the source color of a patch you can select a new color from your image by using the color picker and Shift+click on the patch you want to replace.

Double-click a patch to reset it; Right-click a patch to delete it; Shift+click on empty space to add a new patch (with the currently picked color as the source).

## 8.2.15. color mapping

Transfer the look and feel of one image to another.

This module statistically analyses the color characteristics of a source and target image. The colors of the source image are then mapped to corresponding colors on the target image.

Two steps are required to use this module:

1. Open the source image in the darkroom view and acquire its color characteristics by pressing the "acquire as source" button. A set of color clusters is generated and displayed in the "source clusters" area. Each cluster is represented by a set of color swatches with the mean value in the center surrounded by swatches indicating the variance within that cluster. The clusters are sorted in ascending order by their weight (the number of pixels that contribute to each of them).
2. Open the target image in the darkroom view. The previously collected source clusters should be stored; if they are not yet displayed, press the reset button. Now press the "acquire as target" button to generate a corresponding set of color clusters for your target image, which are then displayed in the "target clusters" area.

When both source and target clusters are collected an automatic color mapping is applied to the target image. In its default settings the overall effect is quite exaggerated. A set of sliders gives you control of the effect's strength. You can also use the *normal* [blend mode](#) along with the *opacity* slider to tame the effect. As the color mapping module comes early in the pipeline, you can also finetune the colors with later modules.

## module controls

### acquire as source/target

Press these buttons to generate color clusters for the source and target image, respectively. The processing takes a few seconds during which the GUI will be unresponsive.

### number of clusters

The number of color clusters to use. This should roughly correlate to the number of dominant colors in the image. The clusters are acquired through a statistical sampling process to represent the colors that occur most commonly throughout the image. The random nature of the sampling process means that you might get different results each time you acquire the clusters, especially if you have a larger number of clusters and/or a complex color palette in the image. If you change this parameter all the collected color clusters are reset and must be re-acquired.

### color dominance

This parameter controls the mapping between source and target clusters. At the minimum value, the mapping is based on color proximity. This typically leads to very subtle effects on the target image. At the maximum value, the mapping is based on the relative weight of the color clusters - dominant colors of the source image are mapped to dominant colors of the target image. This typically leads to a very strong effect. Intermediate values incrementally shift between these extremes.

### histogram equalization

Modify the target image's contrast by matching its histogram with the histogram of the source image. This slider controls the extent of this effect.

## 8.2.16. color reconstruction

Recover color information in blown-out highlights.

Due to the nature of digital sensors, overexposed highlights lack valid color information. Most frequently they appear neutral white or exhibit some color cast, depending on what other image processing steps are involved. This module can be used to “heal” overexposed highlights by replacing their colors with better fitting ones. The module acts on pixels whose luminance exceeds a user-defined threshold. Replacement colors are taken from neighboring pixels. Both the spatial distance and the luminance distance (range) are taken into account for color selection.

Due to a limitation of the underlying algorithm, reconstructed colors may sometimes be displayed incorrectly if you zoom into the image in the darkroom view. If this happens you might observe a magenta shift in highlight areas close to high contrast edges, or you might see colorless highlight areas when used alongside the “reconstruct color” method of the [highlight reconstruction](#) module. These artifacts only influence on-screen image display – the final output remains unaffected. It is recommended that you finetune the parameters of this module only when viewing the fully zoomed-out image.

Note that similar functionality is also available in the *reconstruct* tab of the [filmic rgb](#) module.

### module controls

#### threshold

The color reconstruction module replaces the color of all target pixels having luminance values above this threshold. Only pixels with luminance values below the threshold are taken as valid source pixels for replacement colors. Setting this parameter too high will cause the module to have no effect on any pixels. Setting it too low will minimize the “pool” of replacement colors – if no suitable colors are available, the original colors are retained. This parameter therefore exhibits a “sweet spot” characteristic with an optimum setting depending on the individual image.

#### spatial extent

The spatial distance that source pixels may have from target pixels in order for them to contribute to color replacement. Higher values cause ever more distant pixels to contribute. This increases the chance of finding a replacement color but can make the replacement color less well suited for the reconstruction.

#### range extent

The range difference (in luminance) that source pixels may have from target pixels in order for them to contribute to color replacement. Higher values cause more pixels to contribute even if their luminance differs more strongly from the target pixels. This again increases the chance of finding a replacement color but at the same time increases the risk of unfitting colors creeping in.

#### precedence

This combobox defines whether certain replacement colors shall take precedence over others as follows:

- *off* (default): All pixels contribute equally
- *saturated colors*: Pixels contribute according to their chromaticity – the more highly-saturated a color, the more it contributes
- *hue*: Give precedence to a specific hue (see below)

#### hue

This slider is only visible if you set the *precedence* to “hue”. It allows you to select a preferred hue for replacement colors. This only has an effect if the preferred hue is actually present within the selected spatial and range distance of the target pixels (see above). A typical use case is repairing highlights on human skin in situations where diverging colors are in close proximity (e.g. textiles or hair with a luminance close that of to skin). Setting a hue preference on skin tones prevents these other colors from creeping in.

## 8.2.17. color zones

Selectively adjust the lightness, saturation and hue of pixels based on their current lightness, saturation and hue.

This module works in CIE LCh color space, which separates pixels into *lightness*, chroma (*saturation*) and *hue* components. It allows you to manipulate the lightness, saturation and hue of targeted groups of pixels through the use of [curves](#) .

You first need to choose whether you wish to adjust (select) pixels based on their lightness, saturation or hue. You can then use three curves, on their respective tabs, to adjust the lightness, saturation and hue of ranges of pixels selected via this method.

---

**Note:** This module should be used with care, as transitions between colors may not be graceful. Instead, use the [color balance rgb](#) module with a [parametric mask](#).

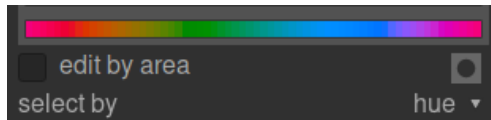
---

## pixel selection method

The *color zones* module offers three different methods for selecting which pixels you want to adjust. They are:

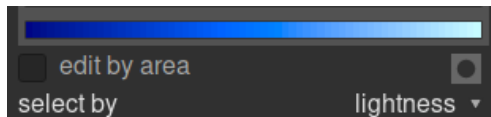
### select by hue (default)

Select pixels to manipulate based on their hue. For example, you may want to darken a blue sky, or to change a red porche into a yellow one. The following image shows the full range of hues that you can choose to operate on:



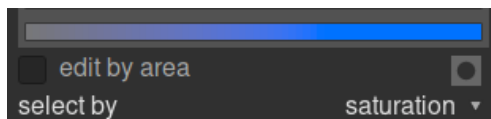
### select by lightness

Select pixels to manipulate based on their lightness. For example, you may want to make your shadows brighter, or to make your highlights more yellow. The following image shows the range of lightness levels that you can choose to operate on, from dark to light:



### select by saturation

Select pixels to manipulate based on their saturation. For example, you may want to tone down the saturation of some already highly saturated pixels, or to change their hue. The following image shows the range of saturation levels that you can choose to operate on, from a completely unsaturated monochrome gray through to the most highly saturated color:



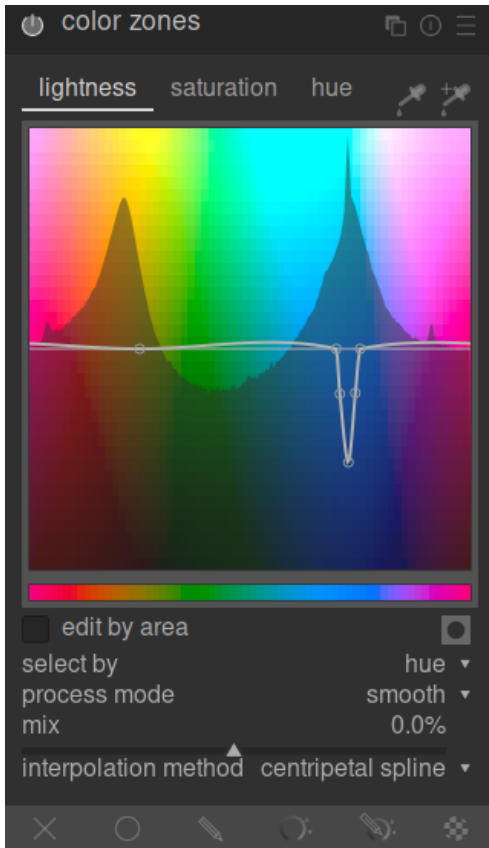
## pixel manipulation curves

Once you have chosen a pixel selection method, the selected range of lightness, saturation or hue levels will appear along the horizontal axis of the three pixel manipulation curves, which can be viewed and adjusted by choosing the appropriate tab (lightness, saturation, hue).

If, for example, you were to choose to *select by hue* (the default), the horizontal axis (below the manipulation curves) would show the range of hues you can work with, and the three pixel manipulation curves would appear as follows:

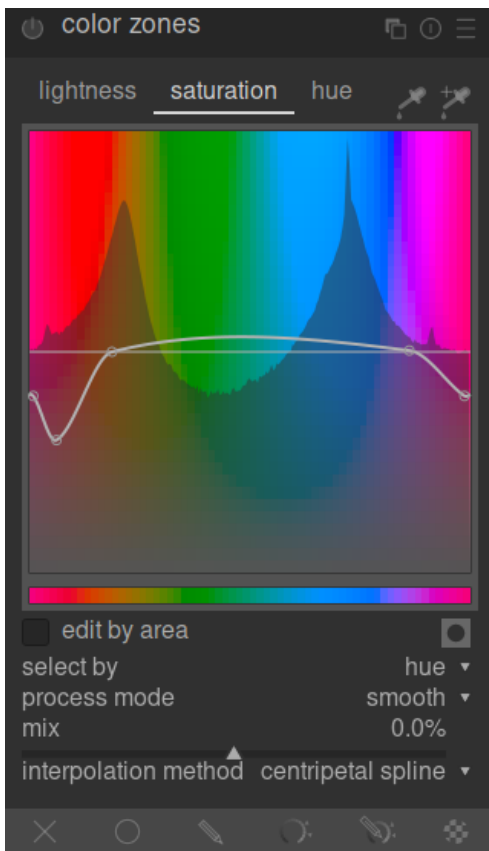
### lightness

By adjusting the lightness curve up or down in a given (hue) location, you can brighten or darken pixels matching hues where the curve has been raised or lowered, respectively. In the example below the blue sky in an image has been darkened for dramatic effect:



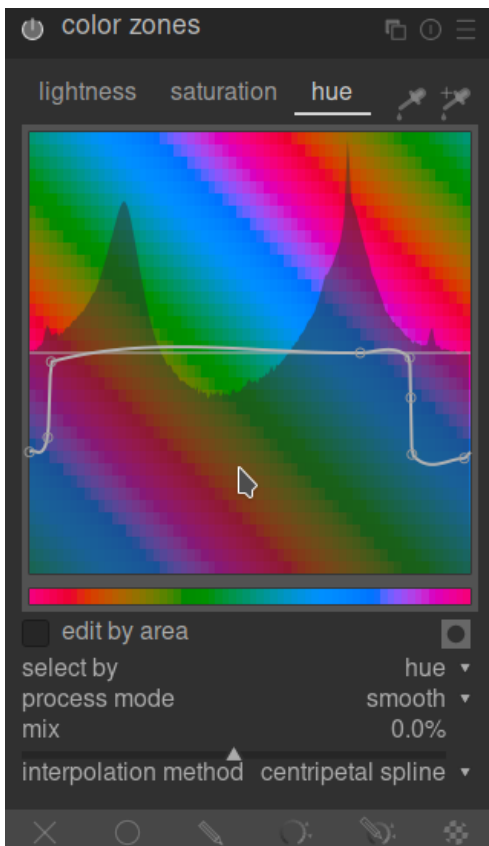
### saturation

By adjusting the saturation curve up or down in a given (hue) location, you can desaturate (make less colorful) or resaturate (make more colorful) pixels matching hues where the curve has been raised or lowered, respectively. In the example below, a red object in the background has been desaturated so it is less of a distraction to the main subject of the photo:



## hue

By adjusting the hue curve up or down in a given (hue) location, you can shift the hue of pixels matching hues where the curve has been raised or lowered, allowing you to replace one color with another. In the example below, a pink toy in an image has been changed to blue:



The curves work similarly in the lightness- and saturation-based selection modes as well. See the section on [curves](#) to see how spline curves work in general.

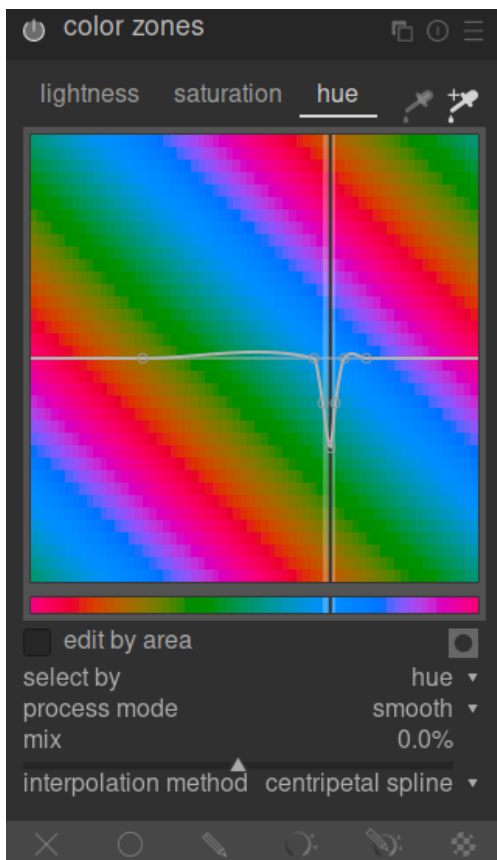
Note that these examples are somewhat contrived in order to illustrate the module's usage. In practical use, they would likely need to be combined with [drawn](#) and/or [parametric](#) masks to further isolate their effect.

## range selection

When adjusting the pixel manipulation curves, it can sometimes be difficult to judge exactly where on the horizontal axis pixels will fall. To the right of the tab controls are a pair of color pickers that can be used to assist with this.

If you click the left-hand color picker and choose a pixel in the image, you will see a dark vertical line showing where that pixel falls on the horizontal axis. If you Ctrl+click or right-click on the same color picker you can choose a rectangular area from the image – the range of values represented within the selected rectangle will be shaded vertically, with a similar dark line showing the median value.

If you click on the right-hand color picker, you can similarly choose a rectangular area on the image and the display will be shown as described above (a shaded area with a dark vertical line). However, in this case the color picker will also automatically add some control points to the curve for you, representing the highlighted range (see below). Simply drag on the center node to raise or lower the curve within the selected range. Alternatively, hold Ctrl while selecting a range to automatically create a positive curve (push up the selected range) or hold Shift while selecting to create a negative curve (pushed down).



## module controls

The following controls are available in the *color zones* module:

### lightness, saturation & hue tabs

Each tab displays a pixel manipulation curve to allow you to alter “lightness”, “saturation”, or “hue” based on the pixel selection method.

**edit by area**

Choose how to interact with the curve. This setting is disabled by default, allowing the control points for the curve to be freely placed. Check the box to fall back to the legacy “edit by area” mode, which functions in a similar way to the spline curve controls used in [wavelet](#) modules.

**mask display**

Enable the *mask display* to highlight pixels that have been affected by *color zones* adjustments in yellow.

**select by**

Define the horizontal axis (the range of values to work on). You can choose between “lightness”, “saturation”, and “hue” (the default). Changing this parameter resets all pixel manipulation curves to their default state (horizontal straight lines). If you want to work on multiple ranges, you need to create additional instances of the module.

**process mode**

Choose between a “smooth” (default) or “strong” processing mode. The default mode is less likely to cause artifacts.

**mix**

Use this parameter to tune the strength of the overall effect.

**interpolation method**

Define how the curve is interpolated using the user-defined control points. See the [curves](#) section for more details.

## 8.2.18. colorize

Add a solid layer of color to the image.

### module controls

**hue**

The hue of the color layer.

**saturation**

The saturation of shadow tones.

**lightness**

The lightness of the color layer.

**source mix**

Controls how the lightness of the input image is mixed with the color layer. Setting this to zero will result in a uniformly colored plane.

## 8.2.19. contrast brightness saturation

A very basic tool for adjusting the *contrast*, *brightness* and *saturation* of an image. Please note that a number of other modules provide much more versatile methods of adjusting these parameters.

All module controls default to a neutral position (zero) and provide the ability to increase or decrease the relevant parameter

---

**Note:** *This module works in Lab color space and is prone to halos. Instead, use the [color balance rgb](#) module.*

---

### module controls

**contrast**

The contrast

**brightness**

The brightness

**saturation**

The saturation

## 8.2.20. contrast equalizer

Adjust luminance and chroma contrast in the wavelet domain.

This versatile module can be used to achieve a variety of effects, including bloom, denoise, clarity, and local contrast enhancement.

It works in the [wavelet](#) domain and its parameters can be tuned independently for each wavelet detail scale. The module operates in CIE LCh color space and so is able to treat luminosity and chromaticity independently.

A number of presets are provided, which should help you to understand the capabilities of the module.

### module controls

The *contrast equalizer* module decomposes the image into various detail scales. On each detail scale, you can independently adjust the contrast and denoise splines for lightness (“luma”) and chromaticity (“chroma”, or color saturation), as well as adjusting the edge-awareness (“edges”) of the wavelet transform. The luma, chroma and edges splines are provided on separate tabs, and some examples of their usage are given in the following sections.

Below the spline graphs is a *mix* slider, which can be used to adjust the strength of the effect, and even to invert the graph (with negative values). While your mouse is over the spline graph, the curve will be displayed as if the *mix* slider was set to 1.0, to allow for easier editing. When you move your mouse away, the graph will be re-adjusted to account for the *mix* slider.

In the background of the curve you can see a number of alternating light and dark stripes. These represent the levels of detail that are visible at your current zoom scale – any details without these stripes are too small to be seen your current view. Adjustments made to control points within the striped section *may* produce a visible effect (depending on the strength of the adjustment). Adjustments outside of the striped region *will not*. Zoom in to see higher levels of detail and make adjustments to the more detailed areas of the image.

---

**Tip:** if you are having trouble visualising which parts of the curve will affect which details in the image, you can set the [blend mode](#) to “difference”. This will make the image go black except for those areas where the output of the module differs from the input. By raising the curve at one of the control points, you will be able to see which details in the image are represented by that point.

---

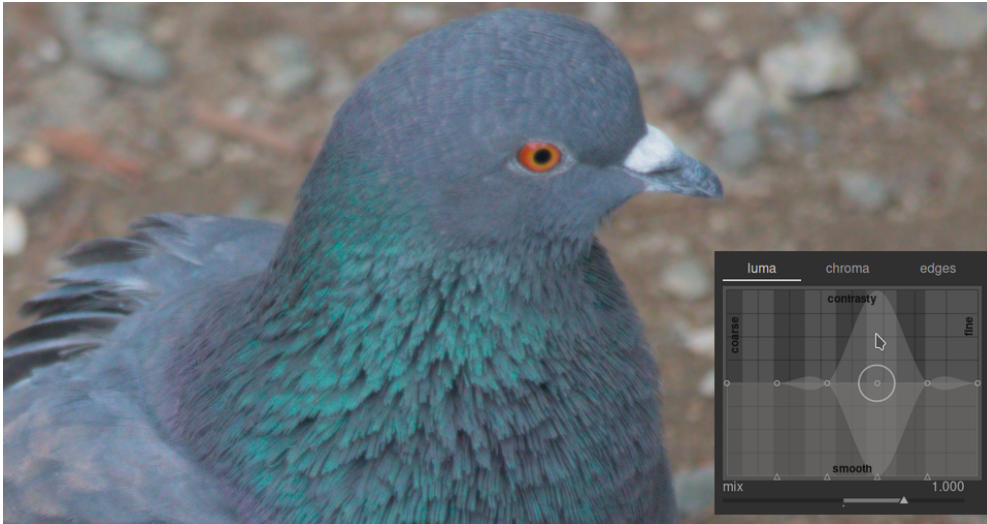
### luma tab

The luma tab allows you to adjust the local contrast in the image’s luminance (brightness). Adjustments are represented by a white spline that begins as a horizontal line running across the centre of the graph (indicating that no change will be made). Raise or lower this spline at the left end of the graph to increase or decrease the local contrast of coarse detail in the image. Perform similar adjustments towards the right side of the graph to adjust the local contrast of the fine details in the image.

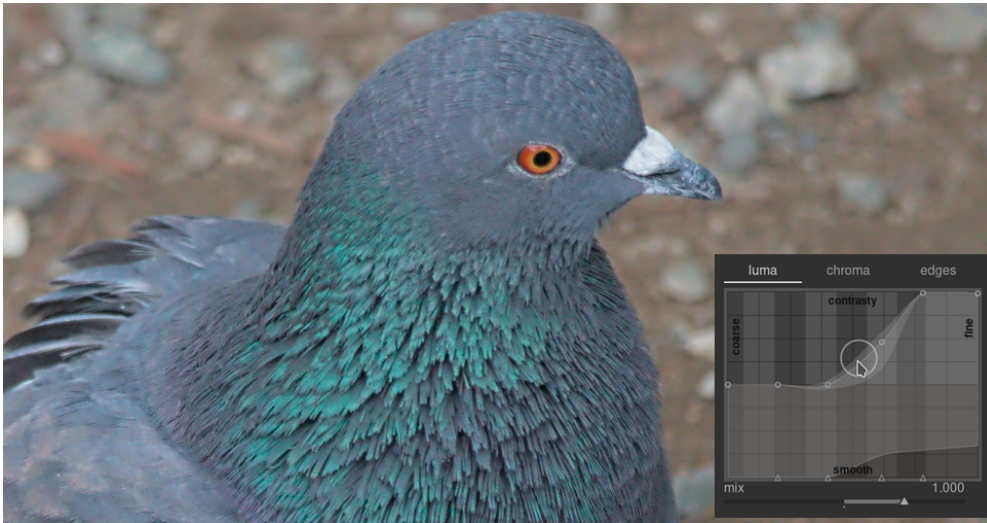
When you hover the mouse pointer over the graph, a white circle indicates the radius of influence of the mouse pointer – the size of this circle can be adjusted by scrolling with the mouse wheel. The larger the circle of influence, the more control points will be affected when you adjust the curve. A highlighted region in the background shows what the spline would look like if you pushed the currently-hovered control point all the way to the top or bottom on the graph – see the screenshot below for examples of these features. For more information see the [wavelets](#) section.

The following image shows the default state of the contrast equalizer module before any adjustments have been made:





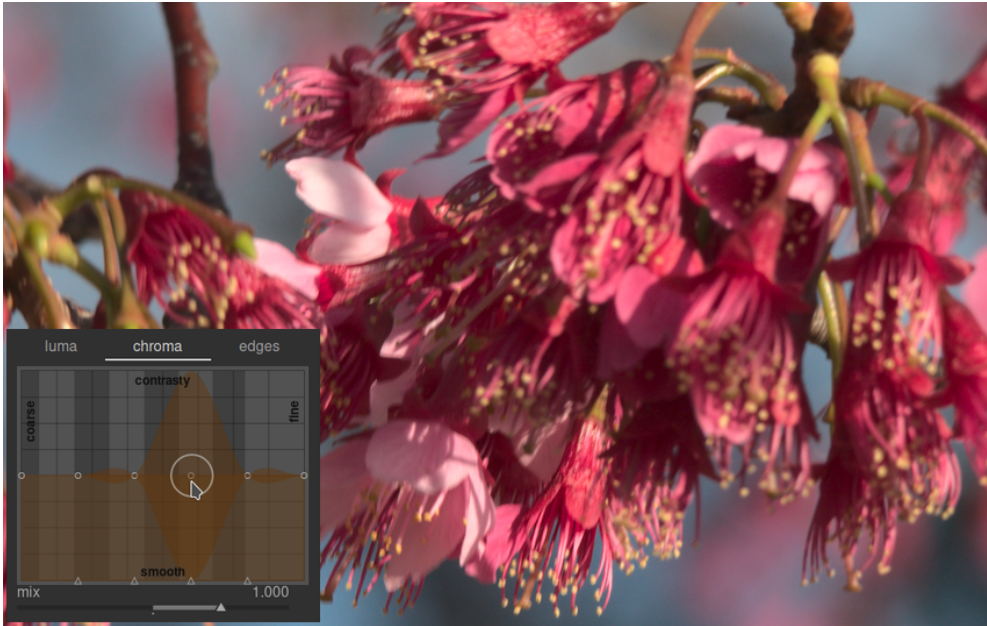
Raising the two control points at the right-hand end of the graph will increase the sharpness of the fine details (the eye and feathers of the bird) while leaving coarser details (the rocks in the background) largely unaffected. The example below has been exaggerated to better illustrate the effect.



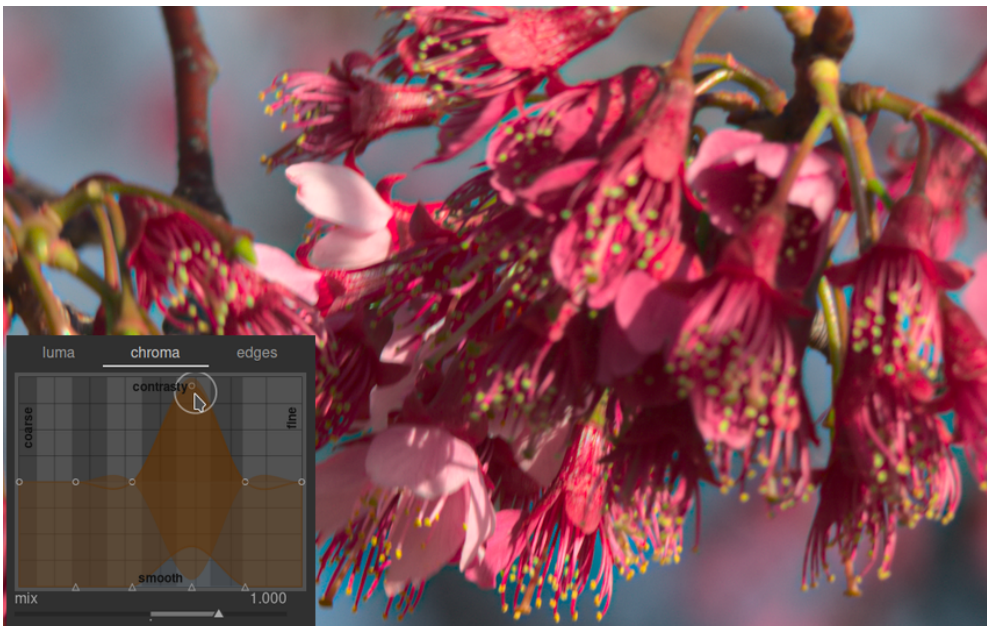
Increasing the local contrast can also amplify the luma noise in the image. A second spline located at the bottom of the graph can be used to denoise the selected detail scales. Raise this spline (by clicking just above one of the triangles at the bottom of the graph and dragging the line upwards) to reduce noise at the given wavelet scale. In the example above, the dark denoising spline has been raised at the fine-detail end of the graph.

## chroma tab

The chroma tab allows the color contrast or saturation to be adjusted at the selected wavelet scales. See the following example:



Say you wanted to bring out the green color of the anthers at the end of the stamen. The pink petals of the flowers are already quite saturated, but using contrast equalizer you can selectively boost the saturation on the small scale of the anthers without impacting the saturation of the petals. By raising the third control point from the right, you can target the saturation of the anthers only:



As in the luma tab, the chroma tab also has a denoising spline at the bottom of the graph. This can be used to handle chroma noise at different scales within the image. Chroma denoising can generally be more aggressive on larger wavelet scales and has less effect on a smaller scale.



## edge tab

The basic wavelet *à trous* transform has been enhanced in the *contrast equalizer* to be “edge-aware”, which can help to reduce the gradient reversals and halo artifacts that the basic algorithm can produce. The *edges* tab does not directly act on the edges in an image; rather it adjusts the edge awareness feature of the wavelet transform. If you have not adjusted the luma or chroma splines, adjusting the edge awareness spline will have no effect.

To see the sorts of artifacts that the *edges* curve tries to combat, here is an example taken from the original paper “Edge-Optimized À-Trous Wavelets for Local Contrast Enhancement with Robust Denoising” (Hanika, Damertz and Lensch 2011):



In the image on the left, the edges spline has been reduced to a minimum, effectively disabling the edge awareness and resulting in halos. In the middle image, the edges spline has been increased too much, resulting in gradient reversals. In the image on the right the edges spline has been set somewhere in between the two extremes, resulting in nice clean edges.

Usually the default central position of the spline is a good starting point, but if there are objectionable artifacts around the edges, this control can be helpful in mitigating them. Some experimentation will be required.

### 8.2.21. crop

Crop an image using on-screen guides.

This module appears later in the pipeline than the deprecated [crop and rotate](#) module, meaning that the full image can remain available for source spots in the [retouch](#) module. For best results, you are advised to use the [rotate and perspective](#) module to perform rotation and perspective correction (if required), and then perform final creative cropping with this module.

Whenever this module is in focus, the full uncropped image will be shown, overlaid with crop handles and optional guiding lines.



Resize the crop by dragging the border and corner handles.

Move the crop rectangle by clicking and dragging inside the crop area. Constrain movement to the horizontal/vertical axis by holding Ctrl/Shift, respectively while dragging. Commit changes by giving focus to another module.

## module controls

The *crop* module controls are split into two sections as follows:

### crop settings

#### aspect

Set the aspect ratio of the crop, constraining the width:height ratio of the crop rectangle to the chosen aspect. Many common numerical ratios are pre-defined. A few special aspect ratios deserve explanation:

- *freehand*: Crop without any restrictions
- *original image*: Retain the aspect ratio of the original image
- *square*: Constrains the aspect ratio to be 1:1
- *golden cut*: The golden ratio (1.62:1)

You can also enter any other ratio after opening the combobox by typing it in the form of “x:y” or as a decimal (e.g. “0.5” to apply a ratio of 2:1).

If you want to add an aspect ratio to the pre-defined drop-down list you can do this by including a line of the form “plugins/darkroom/clipping/extra\_aspect\_ratios/foo=x:y” in darktable’s configuration file \$HOME/.config/darktable/darktable.rc. Here “foo” defines the name of the new aspect ratio and “x” and “y” the corresponding numerical values (x and y must be integers). Note that you can only add new entries for ratios not already present in the drop-down list.

Finally, the button beside the aspect combobox allows you to switch between portrait and landscape orientation if you have selected a rectangular aspect ratio.

---

**Note:** When resizing an image in freehand mode you may retain the currently-set aspect ratio by holding Shift while dragging on any of the resize controls.

---

## margins

These sliders allow you to directly set how much of the image to crop from each side. They are automatically updated if you move or resize the crop area on the image using the mouse.

As this section is rarely used, it is collapsed by default.

### left

The percentage of the image that should be cropped from the left side.

### right

The percentage of the image that should be cropped from the right side.

### top

The percentage of the image that should be cropped from the top.

### bottom

The percentage of the image that should be cropped from the bottom.

## guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.22. crop and rotate (deprecated)

**Please note that this module is [deprecated](#) from darktable 3.8 and should no longer be used for new edits. Please use the [crop](#) module to crop the image, the [rotate and perspective](#) module to perform rotation and keystone correction, and the [orientation](#) module to flip the image on the horizontal/vertical axes.**

Crop, rotate, and correct perspective distortions using on-screen guides.

Whenever the user interface of this module is in focus, the full uncropped image will be shown, overlaid with crop handles and optional guiding lines.





Resize the crop by dragging the border and corner handles.

Move the crop rectangle by clicking and dragging inside the crop area. Constrain movement to the horizontal/vertical axis by holding Ctrl/Shift, respectively while dragging. Commit changes by giving focus to another module.

If you intend to use the [retouch](#) module, you are recommended to use *rotate and perspective* for rotation and/or keystone correction only, performing creative cropping in the [crop](#) module. This process ensures that the entire image is available for source spots in *retouch*, since the *crop* module is placed after *retouch* in the pixelpipe.

---

**Note:** Some of the tools in this module (angle adjustment and perspective distortion correction) require the original image data to be interpolated. For best sharpness results set “lanczos3” as the pixel interpolator in [preferences > processing](#) .

---

## module controls

The *crop and rotate* module controls are split between two tabs as follows:

### main tab

#### flip

Flip the image on the horizontal, vertical or both axes.

#### angle

Correct the rotation angle to level an image by setting a numerical value in degrees or using the mouse. To use the mouse, right-click and drag to draw a line along a suitable horizontal or vertical feature. As soon as you release the mouse button the image is rotated so that the line you drew matches the horizontal or vertical axis.

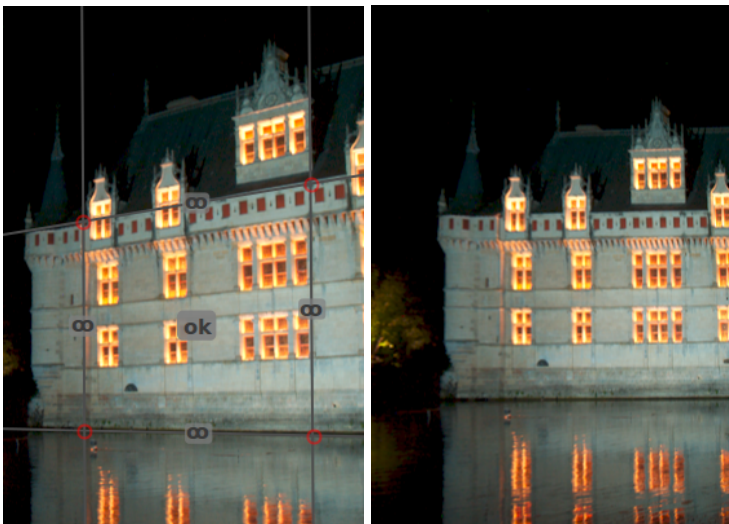
#### keystone

Correct perspective distortions. This tool is useful, for example, when you shoot a high building from the ground with a short focal length, aiming upwards with your camera. The combobox allows you to select the type of correction you want to use:

- *vertical*: Limit the correction to vertical lines
- *horizontal*: Limit the correction to horizontal lines
- *full*: Correct horizontal and vertical lines at the same time

Depending on the selected correction type you will see two or four straight adjustment lines overlaid on your image. Two red circles on each line allow you to modify the line positions with your mouse. Each line additionally carries a “symmetry” button. If activated (and highlighted in red) all movements of the affected line will be mirrored by the opposite line.

In order to correct perspective distortions, you need to find suitable horizontal and/or vertical features in your image and align the adjustment lines with them. When finished, press the “OK” button located close to the center of the image - corrections will be applied immediately. You can return to the module and refine your corrections by selecting “correction applied” in the keystone combobox.



**automatic cropping**

Automatically crop the image to avoid black edges on the image borders. This is useful when rotating the image.

**aspect**

Set the aspect ratio of the crop, constraining the width:height ratio of the crop rectangle to the chosen aspect. Many common numerical ratios are pre-defined. A few special aspect ratios deserve explanation:

- *freehand*: Crop without any restrictions
- *original image*: Retain the aspect ratio of the original image
- *square*: Constrains the aspect ratio to be 1:1
- *golden cut*: The golden ratio (1.62:1)

You can also enter any other ratio after opening the combobox by typing it in the form of “x:y” or as a decimal (e.g. “0.5” to apply a ratio of 2:1).

If you want to add an aspect ratio to the pre-defined drop-down list you can do this by including a line of the form “plugins/darkroom/clipping/extra\_aspect\_ratios/foo=x:y” in darktable’s configuration file \$HOME/.config/darktable/darktable.rc. Here “foo” defines the name of the new aspect ratio and “x” and “y” the corresponding numerical values (x and y must be integers). Note that you can only add new entries for ratios not already present in the drop-down list.

Finally, the button beside the aspect combobox allows you to switch between portrait and landscape orientation if you have selected a rectangular aspect ratio.

---

**Note:** When resizing an image in freehand mode you may retain the currently-set aspect ratio by holding Shift while dragging on any of the resize controls.

---

**show guides**

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

**margins tab**

These sliders allow you to directly set how much of the image to crop from each side. They are automatically updated if you move or resize the crop area on the image using the mouse.

**left**

The percentage of the image that should be cropped from the left side.

**right**

The percentage of the image that should be cropped from the right side.

**top**

The percentage of the image that should be cropped from the top.

**bottom**

The percentage of the image that should be cropped from the bottom.

**8.2.23. defringe (deprecated)**


---

**Please note that this module is [deprecated](#) from darktable 3.6 and should no longer be used for new edits. Please use the [chromatic aberrations](#) module instead.**

---

Remove color fringing, which often results from Longitudinal Chromatic Aberrations (LCA), also known as Axial Chromatic Aberrations.

This module uses edge-detection. Where pixels are detected as a fringe, the color is rebuilt from less saturated neighboring pixels.

## module controls

### operation mode

*global average*: This mode is usually the fastest but might show slightly incorrect previews at high magnification. It might also protect the wrong regions of color too much or too little by comparison with local average.

*local average*: This mode is slower because it computes local color references for every pixel. However it might protect color better than global average and allows for rebuilding color where actually required.

*static threshold*: This mode does not use a color reference but directly uses the threshold as given by the user.

### edge detection radius

The algorithm uses the difference between the input image and a gaussian-blurred version of the same image to detect edges. This parameter controls the spatial extent of the gaussian blur used in the edge detection. Try increasing this value if you want a stronger detection of the fringes or the thickness of the fringe edges is too high.

### threshold

The threshold over which the edge of a pixel is counted as a “fringe”. Try lowering this value if there is not enough fringe detected and increasing it if too many pixels are desaturated.

## 8.2.24. demosaic

Control how raw files are demosaiced.

### bayer filters

The sensor cells of a digital camera are not color-sensitive – they are only able to record different levels of lightness. In order to obtain a color image, each cell is covered by a color filter (red, green or blue) that primarily passes light of that color. This means that each pixel of the raw image only contains information about a single color channel.

Color filters are commonly arranged in a mosaic pattern known as a Bayer filter array. A demosaic algorithm reconstructs the missing color channels by interpolation with data from the neighboring pixels. For further reading see the Wikipedia articles on [demosaicing](#) and the [Bayer filter](#).

Darktable offers several demosaic algorithms, each with its own characteristics. The differences between them are often very subtle and might only be visible while pixel-peeping. However, as the program works on a pixel-by-pixel basis and demosaic generates the base data for the other modules, the choice of the algorithm can have a visually significant effect on the quality of very fine details in the image. This can include the appearance of false moiré patterns as well as the rendering quality of colored edges.

Demosaic interpolation algorithms are often prone to produce artifacts, typically visible as [Moiré patterns](#) when zooming into the image. The chosen algorithm might handle *pre-existing* Moiré- or Maze-like patterns in the raw data in a better or worse way. In these circumstances *VNG4* and *LMMSE* are often more stable.

The following demosaic algorithms are available for sensors with Bayer filters:

- *PPG* used to be darktable’s default demosaic algorithm. It is fast, but other algorithms generally yield better results.
- *AMaZE* and *RCD* offer better reconstruction of high-frequency content (finer details, edges, stars) but might struggle with color reconstruction overshoots or added noise in areas of low contrast. While *AMaZE* often retains more high-frequency details it is also more prone to color overshoots than *RCD*. Since *RCD* now offers similar performance to *PPG*, but with better results, it is now the default algorithm.
- *LMMSE* is better suited for use on high ISO and noisy images than *AMaZE* or *RCD*, both of which tend to generate overshooting artefacts when applied to such images. It can also be useful to manage images that exhibit Moiré patterns with other algorithms.
- *VNG4* is better suited for use on images with low-frequency content (e.g. low contrast regions such as sky) but, compared to *AMaZE* and *RCD*, it often causes loss of some high-frequency details and can sometimes add local color shifts. *VNG* is no longer really recommended – for most images, other available algorithms usually provide better results.

---

**Note:** The performance of the demosaic algorithms differs significantly, *AMaZE* being by far the slowest.

---



## sensors without bayer filters

There are a few cameras whose sensors do not use a Bayer filter. Cameras with an “X-Trans” sensor have their own set of demosaic algorithms. The default algorithm for X-Trans sensors is *Markesteijn 1-pass*, which produces fairly good results. For slightly better quality (at the cost of much slower processing), choose *Markesteijn 3-pass*. Though *VNG* is faster than *Markesteijn 1-pass* on some computers, it is more prone to artifacts.

## special algorithms

*passthrough (monochrome)* is only useful for cameras that have had the color filter array physically removed from the sensor (e.g. scratched off). Demosaic algorithms usually reconstruct missing color channels by interpolation with data from the neighboring pixels. However, if the color filter array is not present, there is nothing to interpolate, so this algorithm simply sets all the color channels to the same value, resulting in a monochrome image. This method avoids the interpolation artifacts that the standard demosaic algorithms might introduce.

*photosite\_color* is not meant to be used for image processing. It takes the raw photosite data and presents it as red, blue or green pixels. This is designed for debugging purposes in order to see the raw data and can assist with analysis of errors produced by the other demosaic algorithms.

## dual demosaic algorithms

Some images have areas best demosaiced using an algorithm that preserves high frequency information (like *AMaZE* or *RCD*) and other areas that might profit from an algorithm more suited to low frequency content (like *VNG4*).

In dual demosaic algorithms (e.g. *RCD + VNG4*) the sensor data is demosaiced twice, first by *RCD*, *AMaZE* or *Markesteijn 3-pass* and then by *VNG4*. Both sets of demosaiced data are retained for subsequent processing.

The data from the high frequency algorithm is then analysed for *local data change* and, using a threshold (there is a bit more of maths involved here), the output image is written pixel-by-pixel for each color channel using data from each demosaic algorithm weighed by the local data change.

In general, areas with greater detail are demosaiced by the algorithm best suited to that purpose (*RCD*, *AMaZe*, *Markesteijn 3-pass*) and any flat areas (like blue sky) are demosaiced using the second algorithm (*VNG4*).

*The ‘local data change’ is technically implemented as a gaussian-blurred single channel selection mask calculated from a combination of the threshold value and the pixels’ luminance.*

## selecting the threshold

An automatically-calculated threshold is difficult to implement. Instead, the “display blending mask” button can be used to display the *selection mask* so you can control the selection of the algorithm manually. The brighter the pixel in the displayed mask, the more the output is taken from the high-frequency algorithm.

## module controls

### method

The demosaic algorithm to use (see above).

### edge threshold (PPG only)

The threshold for an additional median pass. Defaults to “0” which disables median filtering.

### lmmse refine (LMMSE only)

Refinement steps for use with the LMMSE demosaic algorithm. Median steps average the output. Refinement steps add some recalculation of red and blue channels. While the refinement options work well for luma noise, they may decrease quality on images with heavy chroma noise.

### color smoothing

Activate a number of additional color smoothing passes. Defaults to “off”.

### match greens

In some cameras the green filters have slightly varying properties. This parameter adds an additional equalization step to suppress artifacts. Available options are “disabled”, “local average”, “full average” and “full and local average”. This option is not shown for X-Trans sensors.

**switch dual threshold (*dual demosaic modes only*)**

Set the contrast threshold for dual demosaic modes. Lower values favor the high frequency demosaic algorithm and higher values favor the low frequency algorithm.

**display blending mask (*dual demosaic modes only*)**

Show the blending mask that is used to differentiate between high and low frequency areas (adjusted by the “switch dual threshold” parameter). For each pixel, the brighter the mask, the more the module’s output is taken from the high frequency demosaic algorithm.

## 8.2.25. denoise (profiled)

An easy to use and highly efficient denoise module, adapted to the individual noise profiles of a wide range of camera sensors.

One issue with a lot of denoising algorithms is that they assume that the variance of the noise is independent of the luminosity of the signal. By profiling the noise characteristics of a camera’s sensor at different ISO settings, the variance at different luminosities can be assessed, and the denoising algorithm can be adjusted to more evenly smooth out the noise.

Currently, darktable has sensor noise profiles for over 300 popular camera models from all the major manufacturers. If you generate your own noise profile for a camera that is not yet supported by darktable, be sure to share it with the darktable development team so they can include it in the next release! Please see darktable’s [camera support](#) page for more information.

### modes

The *denoise (profiled)* module implements two algorithms, each of which is available in either an easy-to-use “auto” mode, or a more advanced manual mode with additional controls:

**non-local means**

This algorithm works in the spatial domain in much the same way as the [astrophoto denoise](#) module. It averages each pixel with some surrounding pixels in the image. The weight of such a pixel in the averaging process depends on the similarity of its neighborhood with the neighborhood of the pixel being denoised. A patch with a defined size is used to measure that similarity.

Note that this algorithm is quite resource-intensive.

**wavelets (default)**

This algorithm works in the [wavelet](#) domain, and provides a simplified user interface. Wavelet decomposition allows you to adjust the denoise strength depending on the coarseness of the noise in the image. This mode can be used in either *YOUOVO color mode* (which allows you to independently control luminance and chroma noise) or *RGB color mode* (which allows you to independently control noise for each RGB channel).

The wavelet algorithm is less resource-intensive than *non-local means*.

### luma versus chroma noise

Both “non-local means” and “wavelet” algorithms can efficiently tackle luma (lightness) noise and chroma (color) noise.

In the past, it was suggested that you use two separate instances of this module to tackle *chroma* and *luma* noise independently (using *chroma* and *lightness* blending modes). This is no longer recommended, since the *denoise (profiled)* module is placed before the *input color profile* module in the pixelpipe (so that the profile parameters are accurate) and color blending modes should only be used after the input color profile has been applied.

The new algorithms in this module now provide their own methods to separately handle luma and chroma noise, and in both cases this can be handled with a single module instance.

## module controls

The *denoise (profiled)* module provides some controls that are independent of the algorithm used. These are described first, before moving on to the algorithm-specific controls.

When describing the controls specific to an algorithm, we will first cover the simplified interface, and then move on to the more advanced controls for that algorithm.

Note that sliders are provided with minimum and maximum values by default. However, these are only soft limits and, where needed, higher values can be entered by right-clicking on the slider and typing a new value.

## common controls

### profile

darktable automatically determines the camera model and ISO based on the Exif data of your raw file, and searches for a corresponding profile in its database. If your image has an intermediate ISO value, settings will be interpolated between the two closest datasets in the database, and this interpolated setting will show up as the first line in the combo box. You can also manually override this selection if necessary. Re-selecting the top-most entry in the combo box will return you to the default profile.

### mode

Choose which denoising algorithm to use (see above), and whether to present the simplified (“auto”) or full manual interface for that algorithm.

### whitebalance-adaptive transform

As the white balance amplifies each of the RGB channels differently, each channel exhibits different noise levels. This checkbox makes the selected algorithm adapt to the white balance adjustments. This option should be disabled on the second instance if you have applied a first instance with a color blend mode.

### adjust autoset parameters (auto modes only)

Automatically adjust all the other parameters on the current denoising algorithm using a single slider. This is particularly useful when you have had to increase the exposure on an under-exposed image, which normally introduces additional noise (as if you had taken the shot with a higher ISO). This control compensates for that by using settings similar to those used for a higher ISO image. The “effective ISO” used by the denoise algorithm is the actual ISO used, multiplied by the value of this slider.

### strength

Fine-tune the strength of the denoising. The default value has been chosen to maximize the peak signal-to-noise ratio. It’s mostly a matter of taste – whether you prefer a low noise level at the cost of fine details, or you accept more noise to better preserve fine detail.

### preserve shadows (advanced mode only)

Lower this control to denoise the shadows more aggressively. Usually, as noise increases, you will need to decrease this parameter.

### bias correction (advanced mode only)

Correct any color cast that may appear in the shadows. Increase this value if dark shadows appear too green, decrease if they appear too purple.

## non-local means auto sliders

### central pixel weight (details)

Control the amount of detail that should be preserved by the denoising algorithm. By default, this will have a low value, meaning that the algorithm will treat both luma and chroma noise equally. Move this slider to the right to reduce the amount of luma denoising, so that the algorithm will primarily affect chroma noise. By adjusting this slider together with the *strength* slider, you can find a good balance between luma and chroma denoising.

## non-local means advanced sliders

When you take non-local means out of auto mode, the *adjust autoset parameters* slider is replaced with the following controls. You can use the auto-adjust slider to arrive at some initial settings then, when you switch to manual mode, the sliders will show the equivalent manual settings. You can then continue to fine-tune the manual settings from the auto-set starting point.

### patch size

Control the size of the patches being matched when deciding which pixels to average – see [astrophoto denoise](#) for more information. Increase this for images with more noise but be aware that high values may smooth out fine edges. The effect of this slider on processing time is minimal.

### search radius

Control how far away from a pixel the algorithm will attempt to find similar patches. Increasing the value can give better results for very noisy images when coarse grain noise is visible, but the processing time is hugely impacted by this parameter (the processing time increases with the square of this parameter). A lower value will make execution faster, a higher value will make it slower. In most cases it is better to use the scattering parameter, which has a similar effect but without the heavy processing cost.

### scattering (coarse-grain noise)

Like the *search radius*, this slider controls how far from a pixel the algorithm will attempt to find similar patches. However, it does this without increasing the number of patches considered. As such, processing time will stay about the same. Increasing the value will reduce coarse grain noise, but may smooth local contrast. This slider is particularly effective at reducing chroma noise.

## wavelet curves

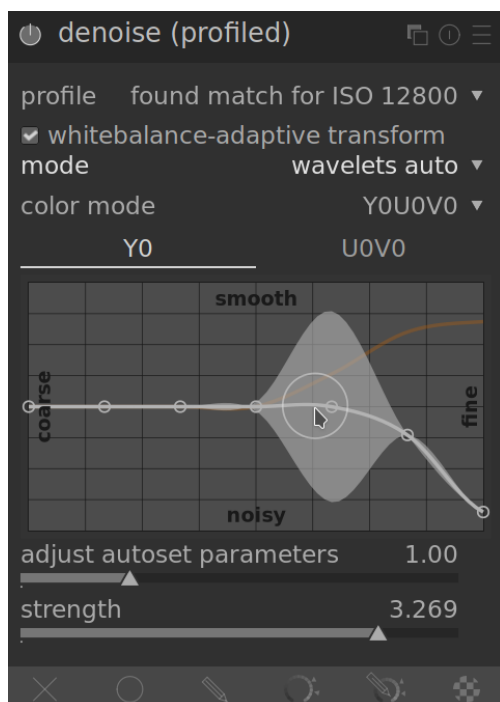
Wavelet curves are shown when one of the “wavelet” modes is selected.

The noise in an image usually consists of both fine grain and coarse grain noise, to varying degrees. The wavelet curves allow the strength of the denoising to be adjusted depending on the coarseness of the visible noise. The left end of the curve will act on very coarse grain noise, while the right of the curve will act on very fine grain noise. Raising the curve will increase the amount of smoothing, while lowering the curve will decrease it.

For example, you can preserve very-fine-grain noise by pulling the right-most part of the curve down. When tackling chroma noise (e.g. on a  $U0V0$  curve, or on a second module instance with a color blend mode) you can safely raise the right side of the curve, since colors do not change a lot on fine scales. This can be useful if you see some noisy isolated pixels with the wrong color.

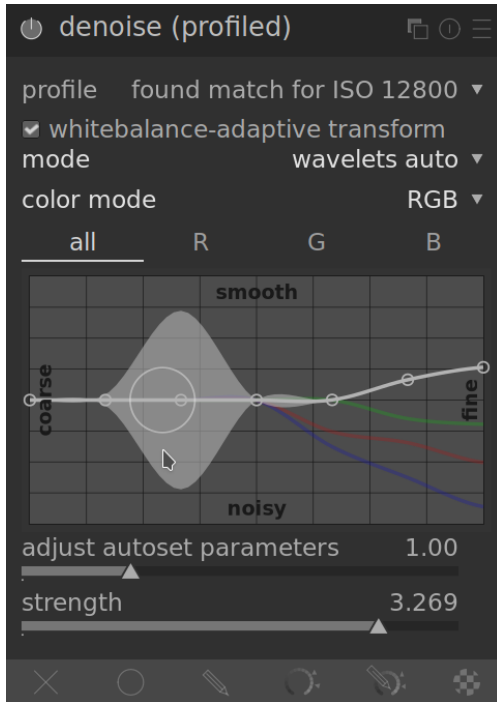
## wavelets $Y0U0V0$ color mode

The preferred way to use wavelets is with the  $Y0U0V0$  color mode. This mode separates the denoising curves into luminance ( $Y0$ ) and color ( $U0V0$ ) components. You can then use the  $Y0$  curve to control the level of luma denoising, and the  $U0V0$  curve to control the level of chroma denoising.



## wavelets *RGB* color mode

Before the *YOU0V0* color mode was introduced, wavelet-based denoising could only be performed directly on the *R*, *G* and *B* channels, either together or individually.



If you want to denoise the RGB channels independently, the best way to do this is to use an instance of the [color calibration](#) module placed immediately before the *denoise (profiled)* module so that it outputs a gray channel based on the *red* channel only, then denoise that monochrome image using the *red* wavelet curve. Repeat this procedure for the blue and green channels. This procedure is time-consuming, but gives the best result because looking at the color of a noisy pixel is not a reliable way to determine which channel to adjust. For example, a noisy red pixel could be due to a noise peak on the red channel, but could also be due to a noise lull on the blue and green channels.

The issue with independently denoising the RGB channels is that there can still be some residual chroma noise at the end that requires excessive smoothing to eliminate. This was in fact one of the key motivations behind implementing the *YOU0V0* color mode.

## wavelets advanced sliders

When you take *wavelets* out of *auto* mode, the *adjust autotset parameters* slider is replaced with the *preserve shadows* and *bias correction* controls listed above in the [common controls](#) section.

## 8.2.26. diffuse or sharpen

Diffusion is a family of physical processes by which particles move and spread gradually with time, from a source that generates them. In image processing, diffusion mostly occurs in two places:

- diffusion of photons through lens glass (blur) or humid air (hazing),
- diffusion of pigments in wet inks or watercolors.

In both cases, diffusion makes the image less sharp by “leaking” particles and smoothing local variations.

The *diffuse or sharpen* module uses a generalized physical model to describe several kinds of diffusion, and can be used by image makers to either simulate or revert diffusion processes.

As it is a highly technical module, several presets are provided to demonstrate its use for various purposes.

Diffusion can be removed in order to:

- recover the original image from sensors with an anti-aliasing filter or mitigate the blur created by most demosaicing algorithms (use the *sharpen sensor demosaicing* preset and move the module before the *input color profile* module in the pipeline),
- reverse static lens blurring/defocusing (use one of the *lens deblur* presets),
- remove atmospheric haze (use the *dehaze* preset),
- add extra acutance for better legibility (use the *local contrast* preset).

Note that motion blurs cannot be reverted by undoing the diffusion process, as they are not diffusive in nature.

Diffusion can be added in order to:

- create a bloom or Orton effect (use the *bloom* preset),
- inpaint missing or damaged parts of an image (use the *inpaint highlights* preset),
- denoise in an edge-preserving way (use one of the *denoise* presets)
- apply a surface blur (use the *surface blur* preset).

Since the process is physical, even its glitches may be used for creative purposes. For example, you can:

- simulate line drawing or watercolor (use the *simulate line drawing* and *simulate watercolor* presets),
- create random patterns and textures by increasing noise (over time, with iterations, noise will connect with neighbours to create random specks).

---

**Note:** This module is highly resource-intensive, as it is actually an anisotropic, multiscale, partial differential equation solver. The module’s runtime increases with the number of iterations and OpenCL is therefore strongly recommended. Some “fast” presets are also provided for use on systems without OpenCL.

---

## concepts

### time

Diffusion is a time-dependent process: the more time it has, the further the particles can spread. In this module, time is simulated using the number of iterations (the number of times the algorithm runs on top of itself). More iterations can make reconstruction (deblurring, denoising, dehazing) more accurate if properly set, but can also cause it to degenerate.

### direction

Natural diffusion usually takes place from points with a high potential (high energy or high concentration of particles) to those with a low potential (low energy or low concentration of particles). In an image, this means that diffusion always occurs from the brightest pixels to the darkest.

This particular implementation can simulate natural diffusion, using what is called an isotropic diffusion (all directions have the same weight, like heat diffusion), but can also force a weighted direction parallel to the gradients (forcing diffusion across object edges and creating ghost edges), or a weighted direction perpendicular to the gradients, called isophote (forcing diffusion to be contained inside edges, like in a droplet of watercolor). The relative weight of each direction (gradient and isophote) is user-defined and can be found in the [diffusion directionality](#) section of the module.

## speed

Depending how fluid the environment is, particles can move more or less freely and therefore more or less fast. The speed of diffusion can be set in the [diffusion speed](#) section of the module.

When performing reconstruction (denoising, deblurring, dehazing), it is advisable to use smaller speeds for better accuracy. This prevents numerical overshoots (and therefore degeneration of the solution) and may require more iterations. For small numbers of iterations, higher speeds may be used. Note that large blurs need many iterations for proper reconstruction, so the speed should be adjusted to avoid degenerating the solution.

All speeds are added (first to fourth orders), and the sums “first order + second order” and “third order + fourth order” should never exceed  $\pm 100\%$ , unless you want to produce [glitch art](#).

## scale

Natural diffusion is supposed to happen only to the closest neighbouring coordinates. That is, at each iteration, each pixel should only interact with its 9 nearest neighbours.

Here, we fast-track things a bit to save time and reuse the multi-scale wavelets scheme from the [contrast equalizer](#) module, so that we can diffuse at different scales. The maximal scale of diffusion is defined by the *radius span* parameter.

Regardless of the diffusion, a *sharpness* parameter allows you to increase or decrease the details at each scale, much like the spline controls of the [contrast equalizer](#). Along with the *edge sensitivity* slider, this provides the same features as the [contrast equalizer](#) module (*luma* and *edges* tabs) but in a scene-referred RGB space.

## module controls

### diffusion properties

#### iterations

The number of times the algorithm should run on top of itself. High values slow the module down but allow more accurate reconstructions, provided that the diffusion speeds are low enough.

#### central radius

The main scale of the diffusion. Zero causes the diffusion to act more heavily on fine details (used for deblurring and denoising). Non-zero values define the size of details to be heavily diffused (used to increase local contrast).

#### radius span

This allows you to select the band of details radii to act on, around the central radius. The span of diffusion defines a range of detail scales (between  $\text{center} - \text{span}$  and  $\text{center} + \text{span}$ ) within which the diffusion is confined. High values diffuse on a large band of radii, at the expense of computation time. Low values diffuse closer around the central radius. If you plan to deblur, the radius span should be approximately the width of your lens blur and the central radius should be zero. If you plan to increase the local contrast, but don't want to affect sharpness or noise, the radius span should be  $3/4$  of your central radius maximum.

The radii are expressed in pixels of the full-resolution image, so copy+pasting settings between images of different resolution may lead to slightly different results, except for pixel-level sharpness.

For electrical engineers, what is set here is a band-pass filter in wavelets space, using a gaussian frequential window centered on *central radius* with a fall-off (standard deviation) of *radius span*. Wavelet scales are analogous to harmonic frequencies and each wavelet scale defines the radius of the details to act on.

### diffusion speed

#### 1st order speed (gradient)

The speed of diffusion of the low-frequency wavelet layers in the direction defined by the *1st order anisotropy* setting. Positive values apply diffusion, negative values undo diffusion, zero does nothing.

#### 2nd order speed (laplacian)

The speed of diffusion of the low-frequency wavelet layers in the direction defined by the *2nd order anisotropy* setting. Positive values apply diffusion, negative values undo diffusion, zero does nothing.

#### 3rd order speed (gradient of laplacian)

The speed of diffusion of the high-frequency wavelet layers in the direction defined by the *3rd order anisotropy* setting. Positive values apply diffusion, negative values undo diffusion, zero does nothing.

### **4th order speed (laplacian of laplacian)**

The speed of diffusion of the high-frequency wavelet layers in the direction defined by the *4th order anisotropy* setting. Positive values apply diffusion, negative values undo diffusion, zero does nothing.

## diffusion directionality

### **1st order anisotropy**

The direction of diffusion of the low-frequency wavelet layers relative to the orientation of the gradient of the low-frequency. Zero is isotropic, negative values make diffusion follow gradients more closely, positive values make diffusion follow isophotes more closely.

### **2nd order anisotropy**

The direction of diffusion of the low-frequency wavelet layers relative to the orientation of the gradient of the high-frequency. Zero is isotropic, negative values make diffusion follow gradients more closely, positive values make diffusion follow isophotes more closely.

### **3rd order anisotropy**

The direction of diffusion of the high-frequency wavelet layers relative to the orientation of the gradient of the low-frequency. Zero is isotropic, negative values make diffusion follow gradients more closely, positive values make diffusion follow isophotes more closely.

### **4rd order anisotropy**

The direction of diffusion of the high-frequency wavelet layers relative to the orientation of the gradient of the high-frequency. Zero is isotropic, negative values make diffusion follow gradients more closely, positive values make diffusion follow isophotes more closely.

## edges management

### **sharpness**

Apply a gain on wavelet details, regardless of diffusion properties. Zero does nothing, positive values sharpen, negative values blur. This is mostly useful as an adjustment variable when blooming or blurring, to retain some sharpness while adding a glow around edges. You are not advised to use this for sharpening alone, since there is nothing to prevent halos or fringes with this setting.

### **edge sensitivity**

Apply a penalty over the diffusion speeds when edges are detected. This detection uses the local variance around each pixel. Zero disables the penalty, higher values make the penalty stronger and more sensitive to edges. Increase if you notice edge artifacts like fringes and halos.

### **edge threshold**

Define a variance threshold, which affects mostly low-variance areas (dark or blurry areas, or flat surfaces). Positive values will increase the penalty for low-variance areas, which is good for sharpening or increasing local contrast without crushing blacks. Negative values will decrease the penalty for low-variance areas, which is good for denoising or blurring with a maximal effect on black and blurry regions.

## diffusion spatiality

### **luminance masking threshold**

This control is useful if you want to in-paint highlights. For values greater than 0%, the diffusion will only occur in regions with a luminance greater than this setting. Note that gaussian noise will be added in these regions to simulate particles and initialize the in-painting.

## workflow

The main difficulty with this module is that while its output can vary dramatically depending on its input parameters, these parameters have no intuitive link to everyday life. Users are likely to be overwhelmed, unless they are already familiar with Fourier partial differential equations. This section proposes some ways to approach this module without the burden of having to understand the underlying theory.



## general advice

If you intend to deblur your image using this module, always start by properly correcting any chromatic aberrations and noise in the image, since the deblurring may magnify these artifacts. It is also important that you don't have clipped black pixels in your image. These can be corrected with the *black level correction* of the [exposure](#) module.

Since it works on separate RGB channels, it is better to apply this module after [color calibration](#), so that you start with a fully neutral, white-balanced, input image. Note that increasing local contrast or sharpness will also lead to a slight color contrast and saturation boost, which is usually a good thing. Since it uses a variance-based regularization to detect edges, it is also better to put this module before any non-linear operation.

## starting with presets

The provided presets have been tuned by the developer and tested on a range of images for typical purposes. The easiest way is simply to start from the presets, and then tweak them as needed:

- if the effect seems too strong, decrease the number of iterations,
- if edge artifacts appear, increase the edge sensitivity,
- if deblurring starts to affect valid blurry parts (bokeh), reduce the radius,
- if deblurring seems correct in bright areas but excessive in dark areas, increase the edges *threshold*,
- if deblurring clips black pixels, lower the *black level correction* in *exposure* module,
- fine-tune the *sharpness* to your taste.

## starting from scratch

The module's default settings are entirely neutral and will do nothing to your image. The spirit of the module is that each order affects the texture of the image in a particular way.

Start by tuning the first order parameters (speed and anisotropy) to get an initial base. Then adjust the radius. This will affect coarser textures (either blur or sharpen them). Remember that the first order acts on the low frequencies of the wavelet scale and follows a direction parallel or perpendicular to the gradient of the low frequencies.

Next, start to tune the second order parameters (speed and anisotropy). The second order also acts on the low frequencies of the wavelet scale but this time follows a direction parallel or perpendicular to the gradient of the *high* frequencies, which can either be the direction of maximal sharpness or of noise. This can be used to reduce noise (using the second order in diffusion mode, with positive values) when you used the first order in sharpening mode (with negative values).

These two steps can be performed on the zoomed-out image. Remember that, while great care has been taken to make the algorithm's visual result fairly scale-invariant, the preview will be exact only when zoomed 1:1. In any case, anything happening at pixel level (radius < 2px) will not be visible for zoom levels lower than 50%.

At this point, you may want to tweak the edge sensitivity to take care of any edge artifacts. In theory, diffusing in the isophote direction ensures that diffusion is contained inside edges, but this is not sufficient when corners and sharp convex shapes are present in the image.

When the the edge sensitivity control has been adjusted to produce satisfying results, the image usually becomes quite soft. In most cases it will be necessary, at this point, to increase the number of iterations in order to compensate. This will come with a performance penalty so tread carefully with the performance/quality trade-off depending on your hardware. If you can't increase the number of iterations, you will have to increase the diffusing speed.

The final step is to fine-tune the third and fourth order, which take care of the high frequencies of each wavelet scale. You will need to be a lot more gentle with these settings than for the first and second orders, as they can cause noise to blow-up really fast.

The third order follows the gradient or isophote direction of the low frequency layer, so can be used to guide the high frequency diffusion in a direction that is more likely to be legitimate regarding real edges (and less prone to catch noise).

The fourth order follows the gradient or isophote direction of the high frequency layer and is more likely to catch noise. Diffusing on the fourth order is the best way to reduce noise without affecting sharpness too much, either as a stand-alone denoiser, or as a regularization step in a deblurring process.

## using multiple instances for image reconstruction

Noise post-filtering may benefit from **introducing** a diffusion process – this can be applied as an extra step after the [denoise \(profiled\)](#) module.

Conversely, the following optical issues may benefit from reconstruction by **undoing** the diffusion process:

1. blur introduced by a sensor's low-pass filter (LPF) and/or anti-aliasing performed by the [demosaic](#) module,
2. static lens blur,
3. haze/fog,
4. light diffusion (using a diffuser that is too large), leading to even lighting and lack of local contrast on the subject.

While more than one of these issues can affect the same picture at the same time, it is better to try to fix them separately using multiple instances of the module. When doing so, ensure the issues are corrected from coarse scale to fine scale, and that denoising always happens first. That is, your instances should appear in the following [pipe order](#) :

1. denoise,
2. local contrast enhancement,
3. dehaze,
4. lens blur correction,
5. sensor and demosaic correction.

Starting with the coarser-scale reconstructions reduces the probability of introducing or increasing noise when performing the finer-scale reconstructions. This is unintuitive because these processes don't happen in this order during the formation of the image. For the same reason, denoising should always happen before any attempt at sharpening or increasing acutance.

## notes and warnings

While this module is designed to be scale-invariant, its output can only be guaranteed at 100% zoom and high quality or full-size export. Results at lower zoom levels or export dimensions may or may not match your expectations.

When setting a deblurring algorithm, try to bear in mind that many of the greatest images in the history of photography were taken with lenses that were not remotely as sharp as those available today. Although the current trend is to build and sell increasingly sharp lenses and have software apply insane amounts of sharpening on top, this fashion does *not* lead to better imagery and makes the retouching process more tedious. Soft focus and a bit of blurriness have some poetic merits too, which surgically-sanitized HD images may fail to convey.

It should be noted that global contrast (using simple tone curves or black/white levels) also affects our perception of sharpness, which is quite different from optical sharpness (optical resolution). Human eyes are only sensitive to local contrast, which may come from optical sharpness (e.g absence of diffusion – thin edges) as well as from amplified tonal transitions. If some global tone mapping is in place to increase the contrast, the image will look sharper. If a tone mapping is used to decrease the contrast, the image will look more blurry. In none of these cases are the actual edges of objects affected in any way, and the perceptual consequences are pure illusion.

Part of the aging process is a loss of eyesight. The amount of sharpening that people over 50 find pleasing may not be the same as for people in their 20s. It is worth considering sharpening to obtain a *plausible* result (matching your everyday perception) rather than a *pleasing* result (that may look good only to people with the same eyesight as yours).

Finally, assessing the sharpness of images zoomed to 1:1 (100%) or more is a foolish task. In museums, exhibitions and even on screen, the general audience looks at images as a whole, not with a magnifying glass. Moreover, in most practical uses, photographs rarely exceed a resolution of 3000×2000 pixels (roughly a 300 DPI print at A4/letter dimensions) which, for 24 Mpx sensors, means downscaling by a factor of 4. When examining a 24 Mpx file at 1:1, you are actually looking at an image that will never exist. Sharpening at pixel level, in this context, is a waste of time and CPU cycles.

## 8.2.27. dithering

This module eliminates some of the banding artifacts that can result when darktable's internal 32-bit floating point data is transferred into discrete 8-bit or 16-bit integer output format for display or export.

Although not an inherent problem in any of darktable's modules, some operations may provoke banding if they produce a lightness gradient in the image. To mitigate possible artifacts you should consider activating dithering when using the [vignetting](#) or [graduated density](#) modules. This is especially relevant for images with extended homogeneous areas such as cloudless sky. Also watch out for banding artifacts when using a gradient [drawn mask](#).

Viewing an image dithered into a very low bit depth from some distance (e.g. "floyd-steinberg 1-bit b&w") will give the impression of a homogeneous grayscale image. darktable attempts to mimic this impression when rendering zoomed-out images in the center view, the navigation window and thumbnails. This is accomplished by dithering those images into a higher number of grayscale levels. Note that, as a consequence, the [scopes](#) module – the data for which is derived from the navigation window – will show this increased number of levels and is therefore not a full match to the output image.

### module controls

#### method

Choose the dithering method to use.

*floyd-steinberg*: Systematically distribute quantization errors over neighboring pixels. This method can be selected with some typical output bit depths. Alternatively, you can select *floyd-steinberg auto*, which automatically adapts to the desired output format.

*random dithering*: This method just adds some level of randomness to break sharp tonal value bands.

#### damping ("random" method only)

Controls the level of added random noise expressed as a damping factor in a  $10^{\log 2}$  basis. A value of -80 is a good fit for 8-bit output formats; -160 for 16-bit output.

## 8.2.28. exposure

Increase or decrease the overall brightness of an image.

This module has two modes of operation:

#### manual

Set the *exposure*, *black level* and *clipping threshold* manually

#### automatic (RAW images only)

Use an analysis of the image's histogram to automatically set the exposure. Darktable automatically selects the exposure compensation that is required to shift the selected *percentile* to the selected *target level* (see definitions below). This mode is particularly useful for automatically altering a large number of images to have the same exposure. A typical use case of automatic mode is deflickering of time-lapse photographs.

### module controls

#### mode

Choose the mode of operation (automatic/manual).

#### compensate camera exposure (manual mode)

Automatically remove the camera exposure bias (taken from the image's Exif data).

#### exposure (manual mode)

Increase (move to the right) or decrease (move to the left) the exposure value (EV).

#### clipping threshold (manual mode)

Define what percentage of bright values are to be clipped in the calculation of the *exposure* and *black level correction*.

Use the color picker to sample a portion of the image to be used for this calculation.

#### percentile (automatic mode)

Define a location in the histogram to use for automatic exposure correction. A percentile of 50% denotes a position in the histogram where 50% of pixel values are above and 50% of pixel values are below that exposure.

**target level (automatic mode)**

Define the target level for automatic exposure correction (EV) relative to the white point of the camera.

**black level correction (manual and automatic modes)**

Adjust the black level point to unclip negative RGB values.

---

**Note:** Do not use the black level correction to add more density in blacks as this can clip near-black colors out of gamut by generating negative RGB values. This can cause problems with some modules later in the pixelpipe. Instead, use a tone mapping curve to add density to the blacks. For example, you can use the *relative black exposure* slider on the *scene* tab of the [filmic rgb](#) module, or establish a deeper toe in the [base curve](#) module.

---

## 8.2.29. fill light (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [tone equalizer](#) module or the [exposure](#) module with a [drawn mask](#) .**

---

Locally modify exposure based on pixel lightness.

This module pushes exposure by increasing lightness with a Gaussian curve of a specified width, centered on a given lightness.

### module controls

**exposure**

The fill-light exposure (EV)

**center**

The median lightness impacted by the fill-light. The lightness must be set manually but a color picker can be used to provide a guide based on a sample from the image. The lightness of the selected point/area is displayed in the gradient bar.

**width**

The width of the Gaussian curve. This number is expressed in zones, with the full range being 10 zones.

## 8.2.30. filmic rgb

Remap the tonal range of an image by reproducing the tone and color response of classic film.

This module can be used either to expand or to contract the dynamic range of the scene to fit the dynamic range of the display. It protects colors and contrast in the mid-tones, recovers the shadows, and compresses bright highlights and dark shadows. Highlights will need extra care when details need to be preserved (e.g. clouds).

The module is derived from another module of the same name in [Blender 3D modeller](#) by T. J. Sobotka. While it is primarily intended to recover high-dynamic-range images from raw sensor data it can be used with any image in place of the [base curve](#) module. The following video (by the developer of this module) provides a useful introduction: [filmic rgb: remap any dynamic range in darktable 3](#) .

*filmic rgb* is the successor to the *filmic* module from darktable 2.6. While the underlying principles have not changed much, the default settings and their assumptions have, so users of the previous version should not expect a 1:1 translation of their workflow to the new version.

---

**Note:** Despite the technical look of this module, the best way to set it up is to assess the quality of the visual result. Do not overthink the numbers that are presented in the GUI to quantify the strength of the effects.

---

### prerequisites

In order to get the best from this module, your images need some preparation:

---

### capturing (ETTR)

In-camera, it is recommended that you use a technique known as “Expose To The Right” (ETTR). This means exposing the shot so that the exposure is as bright as possible without clipping the highlights. It is called “exposing to the right” because the in-camera histogram should be touching all the way up to the right hand side without peaking at the right hand side (which could indicate clipping). This technique ensures you make maximum use of the dynamic range of your camera’s sensor.

The default auto-exposure metering mode in your camera will normally expose the image so that the average brightness in the image tends towards middle-gray. Sometimes, for scenes dominated by light tones, the camera will underexpose the image to bring those light tones more towards middle-gray. For scenes dominated by dark tones, it may over-expose the image and end up clipping the highlights. In such cases you can use the exposure compensation dial in your camera to raise or lower the exposure – the darktable exposure module can automatically take this into account when processing your image.

In some cases (e.g. specular highlights reflecting off shiny objects) it may be acceptable to have some clipping, but be aware that any clipped data in your image is irrevocably lost. Where data has been clipped, *filmic rgb* offers a “highlight reconstruction” feature to help mitigate the effects of the clipping and blend it smoothly with the rest of the image. The settings for this feature are on the [reconstruct](#) tab. Some cameras also offer a “highlight priority” exposure metering mode that can help to maximise exposure while protecting the highlights, and many offer features such as “zebras” or “blinkies” in the live view to alert the photographer when parts of the image are being clipped.

### adjust for the mid-tones

In the [exposure](#) module, adjust the exposure until the mid-tones are clear enough. Don’t worry about losing the highlights at this point – they will be recovered as part of the filmic processing. However, it is important to avoid negative pixels in black areas else the computations performed by *filmic rgb* may produce unpredictable results. For some camera models (Canon, mainly), rawspeed (the raw decoding library of darktable) may set an exaggerated black level, resulting in crushed blacks and negative pixel values. If so, brighten the blacks by setting a negative black level correction value in the [exposure](#) module.

### white balance, denoise, demosaic

If you plan on using *filmic rgb*’s auto-tuners, use the [white balance](#) module to first correct any color casts and obtain neutral colors. In RGB color spaces, luminance and chrominance are linked, and *filmic rgb*’s luminance detection relies on accurate measurements of both. If your image is very noisy, add an initial step of denoising to improve the black exposure readings, and use a high quality [demosaic](#) algorithm. You don’t need to worry about noise if you are planning to set up filmic manually, without using the auto-tuners.

### disable tone mapping modules

If you plan to use one of *filmic rgb*’s chrominance preservation modes, avoid using [base curve](#) and the various tone mapping modules. These may produce unpredictable color shifts that would make the chrominance preservation useless. None of these modules should be required when using *filmic rgb*.

## usage

The *filmic rgb* module is designed to map the dynamic range of the photographed scene (RAW image) to the dynamic range of the display.

This mapping is defined in three steps, each handled in a separate tab in the interface:

- The [scene](#) tab contains the “input” settings of the scene, defining what constitutes white and black in the photographed scene.
- The [reconstruct](#) tab offers tools to handle blown highlights.
- The [look](#) tab contains the artistic intent of the mapping that is applied to the input parameters (as defined in the scene tab). This part of the module applies an S-shaped parametric curve to enhance the contrast of the mid-tones and remap the gray value to the middle-gray of the display. This is similar to what the [base curve](#) and [tone curve](#) modules do. As a general guideline, you should aim to increase the latitude as much as possible without clipping the extremes of the curve.
- The [display](#) tab defines the output settings required to map the transformed image to the display. In typical use cases, the parameters in this tab rarely require adjustment.
- The [options](#) tab includes some optional advanced settings and parameters.

*filmic rgb* tends to compress local contrast, so after you have finished adjusting settings here, you may wish to compensate for this using the [local contrast](#) module. You may also want to increase the saturation in the [color balance rgb](#) module, and perhaps to further adjust the tones using the [tone equalizer](#).

The ranges of *filmic rgb*'s sliders are limited to typical and safe values, but you can enter values outside of these limits by right-clicking and entering values with the keyboard.

---



**Note:** *filmic rgb* cannot be set with entirely neutral parameters (resulting in a “no-operation”) – as soon as the module is enabled, the image is always at least slightly affected. You can, however, come close to neutral with the following settings:

- in the [look](#) tab, set contrast to 1.0, latitude to 99 % and mid-tones saturation to 0 %,
- in the [options](#) tab, set contrast in shadows and in highlights to *soft*.

In this configuration, filmic will only perform a logarithmic tone mapping between the bounds set in the [scene](#) tab.

---

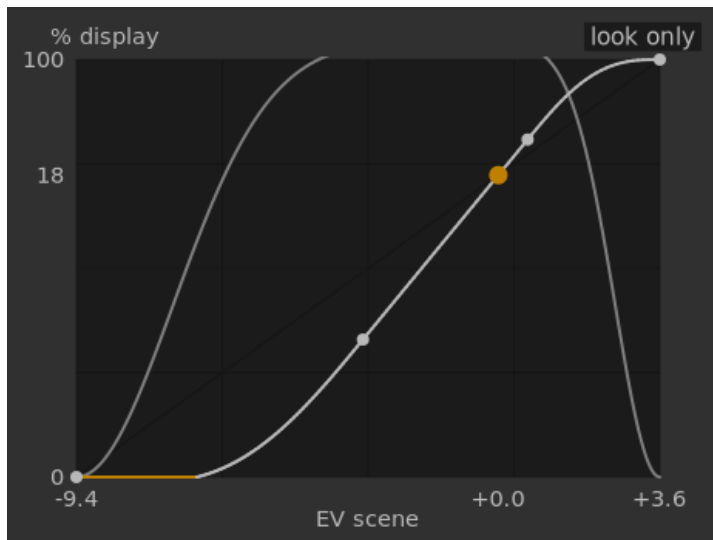
## graphic display

The graphic display at the top of the *filmic rgb* module offers multiple views to help you to understand its functionality. You can cycle through these views using the  icon to the right of the graph display. You can also toggle the labels on the axes on and off using the  icon.

The following views are available:

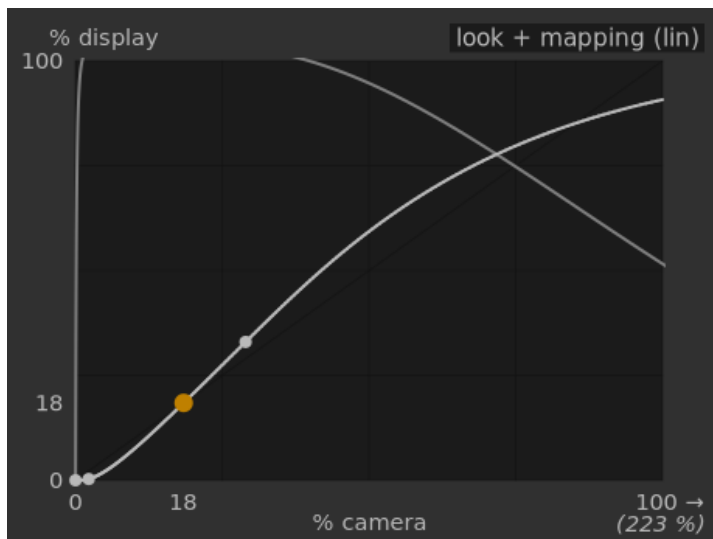
### look only

This is the default view. The main bright curve shows how the dynamic range of the scene (in EV) is compressed into the display-referred output range. The orange dot shows the middle-gray point, the white dots either side mark out the latitude range, and the orange part of the curve at the bottom indicates an overshoot problem with the spline (the [look](#) tab has some controls to deal with this). The darker curve shows how the color saturation rolls off in the extreme shadows and highlights.



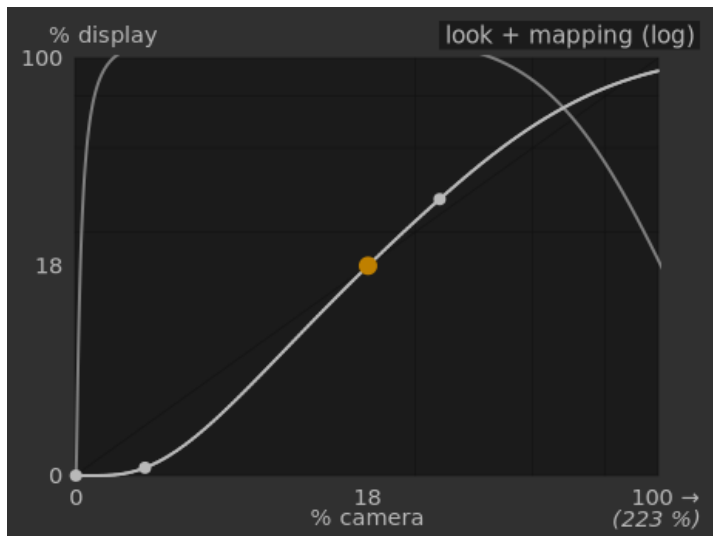
### look + mapping (linear)

This view shows the mapping of input values [0,1] to output values in linear space, including the dynamic range mapping and the output transfer function. Note that in a scene-referred workflow, input values are allowed to exceed 1, however the graph only shows in/out values in the interval [0,1] in order to make the shape of the graph comparable to other tone curve mapping tools such as *base curve* or *tone curve*. The actual value of the scene white point is shown in brackets on the X axis (expressed as a percentage of an input value of 1).



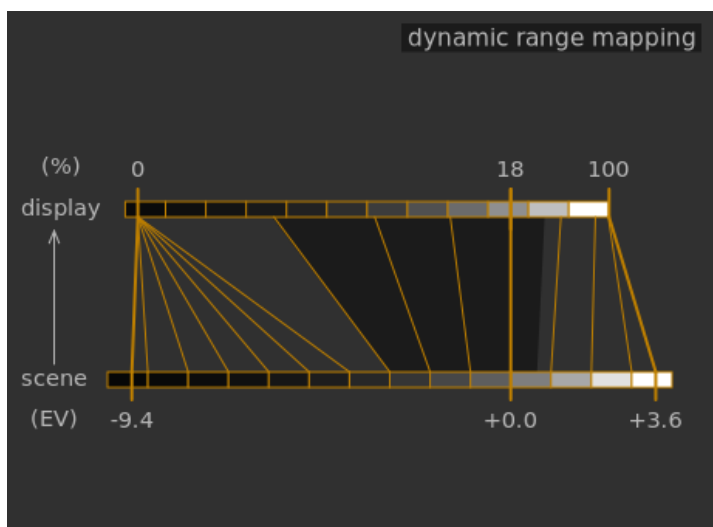
### look + mapping (log)

The same as the previous view, but plotted in log space.



### dynamic range mapping

This view is inspired by the Ansel Adams Zone System, showing how the zones in the input scene (EV) are mapped to the output. Middle gray from the scene is always mapped to 18% in the output (linear) space, and the view shows how the tonal ranges towards the extremes of the scene exposure range are compressed into a smaller number of zones in the display space, leaving more room for the mid-tones to be spread out over the remaining zones. The latitude range is represented by the darker gray portion in the middle.



**Note:** When some parameters are too extreme, resulting in an unfeasible curve, *filmic rgb* will sanitize them internally. Sanitizing is illustrated in two ways on the look views:

- A dot becoming red indicates that the linear part of the curve is pushed too far towards the top or the bottom. In the [look](#) tab, reduce the *latitude* or recenter the linear part using the *shadows* ↔ *highlights balance* parameter.
- A dot becoming a half circle indicates that contrast is too low given the dynamic range of the image. Increase *contrast* in the [look](#) tab, or the *dynamic range* in the [scene](#) tab.

## module controls

### scene

The controls in the *scene* tab are similar in principle to those of the [levels](#) module (black, gray, white). The difference is that *levels* assumes display-referred pixels values (between 0 and 100%), whereas *filmic* allows you to work on scene-referred pixels (between -infinity EV and +infinity EV), which forces the use of a different interface.



**middle-gray luminance (hidden by default)**

This setting allows you to decide what luminance in the scene should be considered the reference middle-gray (which will be remapped to 18% in display). Use the color picker tool to read the average luminance over the drawn area. If you have a photograph of a gray card or a color chart (IT8 chart or colorchecker) shot in the scene lighting conditions, then the gray color picker tool can be used to quickly sample the luminance of the gray patch on that image. In other situations, the color picker can be used to sample the average luminance of the subject.

This has an effect on the picture that is analogous to a brightness correction. Values close to 100% do not compress the highlights but fail to recover shadows. Values close to 0% greatly recover the shadows but compress the highlights more harshly and result in local-contrast losses.

When modifying the middle-gray luminance, the white and black exposures are automatically adjusted accordingly, to prevent the dynamic range from clipping and to help you set the right parameter faster. If you are not happy with the auto adjustment performed by the gray slider, you can correct the white and black exposure parameters afterwards.

---

**Note:** You are not advised to use this control to set middle-gray, hence it is now hidden by default. You should instead use the *exposure* module to set the middle-gray level (see [usage](#), above). However, if you wish to make this slider visible, you can enable it with the *use custom mid-gray values* checkbox in the [options](#) tab.

---

**white relative exposure**

The number of stops (EV) between the scene middle-gray luminance and the scene luminance to be remapped to display white (peak-white). This is the right bound of the scene dynamic range that will be represented on the display - everything brighter than this value on the scene will be clipped (pure white) on the display. The color picker tool reads the maximum luminance in RGB space over the drawn area, assumes it is pure white, and sets the white exposure parameter to remap the maximum to 100% luminance.

**black relative exposure**

The number of stops (EV) between the scene middle-gray luminance and the scene luminance to be remapped to display black (maximum density). This is the left bound of the scene dynamic range that will be represented on the display - everything darker than this value on the scene will be clipped (pure black) on the display. The color picker tool reads the minimum luminance in RGB space over the drawn area, assumes it is pure black, and sets the black exposure parameter to remap the minimum to 0% luminance. The black color picker measurement is very sensitive to noise, and cannot identify whether the minimum luminance is pure black (actual data) or just noise. It works better on low ISO pictures and with high quality demosaicing. When the color picker puts the black exposure at -16 EV, this is a sign that the measurement has failed and you will need to adjust it manually.

The black relative exposure allows you to choose how far you want to recover lowlights.

**dynamic range scaling and auto-tune**

The auto-tune color picker combines the above color pickers, and allows you to set the white and black exposures at the same time, using the maximum of the drawn region as the white and the minimum as the black. This gives good results in landscape photography but usually fails for portraits and indoor scenes.

When no true white and black are available on the scene, the maximum and minimum RGB values read on the image are not valid assumptions any more. Dynamic range scaling symmetrically shrinks or enlarges the detected dynamic range and the current parameters. This works with both color pickers, and adjusts the current values of white and black relative exposures.

---

**Note:** There is no direct relationship between your camera sensor's dynamic range (to be found in DxoMark.com or PhotonsToPhotos.org measurements) and the dynamic range in filmic (scene white EV - scene black EV). Many things happen before *filmic* in the pipeline (for example a black raw offset that could map black to 0) such that *filmic* sees a theoretically infinite dynamic range at its input. This has to do only with pixel encoding manipulation in software, not actual sensor capabilities.

The *scene-referred* workflow forces a black level correction of -0.0002, in the *exposure* module, which ensures that the dynamic range seen by *filmic*'s input is around 12.3 EV most of the time. Decrease this value even more if setting the black relative exposure in *filmic* to -16 EV fails to unclip blacks.

---

## reconstruct

This tab provides controls that blend transitions between unclipped and clipped areas within an image and can also help to reconstruct colors from adjacent pixels. It is designed to handle spotlights that could not possibly be unclipped when taking the shot (such as naked light bulbs or the sun disc in the frame) and aims at diffusing their edges as film would do. It is not designed to recover large areas of clipped pixels or in-paint missing parts of the image.

It can sometimes be useful to disable the [highlight reconstruction](#) module in order to provide additional data to the reconstruction algorithm (*highlight reconstruction* clips highlight data by default). You should note that this can lead to magenta highlights, which will need to be handled with the *gray/colorful details* slider.

Firstly, a mask needs to be set up to identify the parts of the image that will be affected by the highlights reconstruction. There are then some additional controls to fine-tune some of the trade-offs made by the reconstruction algorithm.

### *highlights clipping*

These controls allow you to choose which areas of the image are impacted by the highlight reconstruction algorithms.

#### **threshold**

Any pixels brighter than this threshold will be affected by the reconstruction algorithm. The units are in EV, relative to the white point set in the *scene* tab. By default, this control is set to +3 EV, meaning that pixels need to be at least +3 EV brighter than the white point set in the [scene tab](#) in order for the highlight reconstruction to have any effect. In practise, this means that highlight reconstruction is effectively disabled by default (for performance reasons - it should only be enabled when required). Therefore, to use the *highlights reconstruction* feature, first click the *display highlight reconstruction mask* icon to show the mask, and lower this threshold until the highlight areas you want to reconstruct are selected in white by the mask. It may be useful to first review the image using the [raw overexposed warning](#) to show you which pixels in the raw file have been clipped, and whether those pixels are clipped on just one RGB channel or all of them.

#### **transition**

Use this control to soften the transition between clipped and valid pixels. Moving this control to the right will increase the amount of blur in the mask, so that the transition between clipped and non-clipped areas is softer. This allows for a smoother blending between the clipped and non-clipped regions. Moving this control to the left will reduce the blur in the mask, making the transition in the mask much sharper and therefore reducing the amount of feathering between clipped and non-clipped areas.

#### **display highlight reconstruction mask**

Click on the icon to the right of this label to toggle the display of the highlight reconstruction mask. It is recommended that you turn this on while adjusting the above controls.

### *balance*

These controls allow you to balance the trade-offs between the various reconstruction algorithms.

#### **structure ↔ texture**

Use this to control whether the reconstruction algorithm should favor painting in a smooth color gradient (structure), or trying to reconstruct the texture using sharp details extracted from unclipped pixel data (texture). By default, the control is in the middle at 0%, which favors both strategies equally. If you have lots of areas where all three channels are clipped, there is no texture detail available to reconstruct, so it is better to move the slider to the left to favor color reconstruction. If you have lots of areas where only one or two channels are clipped, then there may be some texture detail in the unclipped channel(s), and moving the slider to the right will place more emphasis on trying to reconstruct texture using this unclipped data.

#### **bloom ↔ reconstruct**

Use this to control whether the algorithm tries to reconstruct sharp detail in the clipped areas (reconstruct), or apply a blur that approximates the blooming effect you get with traditional film (bloom). By default, this is set to 100%, which tries to maximise the sharpness of the detail in the clipped areas. Move this slider to the left if you want to introduce more blur in these areas. Introducing more blur will usually tend to darken the highlights as a by-product, which may lead to a more colorful reconstruction.

#### **gray ↔ colorful details**

Use this to control whether the algorithm favors the recovery of monochromatic highlights (gray) or colorful details. Move the slider to the right if you want more color in the highlights. Move the slider to the left if you want to reduce the saturation of the highlights. It can be helpful to reduce the saturation in the highlights if you see start seeing magenta or out-of-gamut colors.

## look

When working on the *look* tab, it is recommended that you monitor the S-curve spline on the *look only* graph. This curve starts from the scene/display black levels at the bottom left of the graph, and should smoothly increase up to the scene/display white levels at the top right. Sometimes, if the constraints on the S-curve are too tight, the splines in the shadows and/or highlights regions can “overshoot” the limits of the display, and an orange warning is shown on those parts of the spline.

If you see the orange warning indicator at either end of the S-curve, corrective actions should be performed to bring the S-curve back to a smooth monotonically increasing curve. This may involve:

- reducing the latitude and/or contrast,
- adjusting the shadows/highlights slider to shift the latitude and allow more room for the spline,
- ensuring that the scene-referred black and white relative exposure sliders on the *scene* tab have been properly set for the characteristics of the scene,
- setting one or both of the contrast settings on the [options](#) tab to *safe* or *hard*.

If the *target black luminance* setting on the [display](#) tab is non-zero, this can also make it difficult for *filmic rgb* to find a smooth monotonic spline, and reducing this can also help to relax the constraints. See the [display](#) section to understand the implications of this.

### contrast

The *filmic* S-curve is created by computing the position of virtual nodes from the module parameters and interpolating them. This is similar to how the tone curve module operates, but here, the nodes cannot be moved manually. The curve is split into three parts – a middle linear part, and two extremities that transition smoothly from the slope of the middle part to the ends of the exposure range.

The contrast slider controls the slope of the middle part of the curve, as illustrated in the graph display. The larger the dynamic range, the greater the contrast should be set to, in order preserve a natural-looking image. This parameter mostly affects the mid-tones. Note that global contrast has an impact on the acutance (perceived sharpness) – a low-contrast image will look unsharp even though it is optically sharp in the sense of the [Optical Transfer Function \(OTF\)](#).

Setting the contrast to 1 almost completely disables the S-curve, though there will be a very small residual effect from the splines in the highlights and shadows.

### hardness (previously target power factor function)

Known as the *target power factor function* slider in older versions of *filmic rgb*, this slider is hidden by default, and is adjusted automatically based on values in the [scene](#) tab. To make this slider visible, you need to uncheck *auto adjust hardness* in the [options](#) tab.

This parameter is the power function applied to the output transfer function, and it is often improperly called the *gamma* (which can mean too many things in imaging applications, so we should stop using that term). It is used to raise or compress the mid-tones to account for display non-linearities or to avoid quantization artifacts when encoding in 8 bit file formats. This is a common operation when applying ICC color profiles (except for linear RGB spaces, like REC 709 or REC 2020, which have a linear “gamma” of 1.0). However, at the output of *filmic rgb*, the signal is logarithmically encoded, which is not something ICC color profiles know to handle. As a consequence, if we let them apply a gamma of 1/2.2 on top, it will result in a double-up, which would cause the middle-gray to be remapped to 76% instead of 45% as it should in display-referred space.

### latitude

The latitude is the range between the two nodes enclosing the central linear portion of the curve, expressed as a percentage of the dynamic range defined in the [scene](#) tab (white relative exposure minus black relative exposure). It is the luminance range that is remapped in priority, and it is remapped to the luminance interval defined by the contrast parameter. It is usually advisable to keep the latitude as large as possible, while avoiding clipping. If clipping is observed, you can compensate by either decreasing the latitude, shifting the latitude interval with the *shadow ↔ highlights balance* parameter, or decreasing the contrast.

The latitude also defines the range of luminances that are not desaturated at the extremities of the luminance range (See *mid-tones saturation*).

### shadows ↔ highlights balance

By default, the latitude is centered in the middle of the dynamic range. If this produces clipping at one end of the curve, the balance parameter allows you to slide the latitude along the slope, towards the shadows or towards the highlights. This allows more room to be given to one extremity of the dynamic range than to the other, if the properties of the image demand it.

### mid-tones saturation (previously *extreme luminance saturation*)

At extreme luminances, the pixels will tend towards either white or black. Because neither white nor black have color associated with them, the saturation of these pixels must be 0%. In order to gracefully transition towards this 0% saturation point, pixels outside the mid-tone latitude range are progressively desaturated as they approach the extremes. The darker curve in the *filmic rgb* graph indicates the amount of desaturation that is applied to pixels outside the latitude range. Moving the slider to the right pushes the point where desaturation will start to be applied towards the extremes, resulting in a steeper desaturation curve. If pushed too far, this can result in fringing around the highlights. Moving the slider to the left brings the point at which color desaturation will start to be applied closer to the center, resulting in a gentler desaturation curve. If you would like to see more color saturation in the highlights, and you have checked that the white relative exposure in the [scene](#) tab is not yet clipping those highlights, move the mid-tones saturation slider to the right to increase the saturation.

Please note that this desaturation strategy has changed compared to previous versions of *filmic rgb* (which provided a different slider control labelled *extreme luminance saturation*). You can revert to the previous desaturation behaviour by selecting “v3 (2019)” in the *color science* setting on the [options](#) tab.

This control is set to 0 by default and it is now recommended that saturation is handled earlier in the pipeline. A preset “add basic colorfulness” has been added to the [color balance rgb](#) module for this purpose.

## display

The parameters in this tab should rarely require adjustment.

### target black luminance

The destination parameters set the target luminance values used to remap the tones. The default parameters should work 99% of the time, the remaining 1% being when you output in linear RGB space (REC709, REC2020) for media handling log-encoded data. These settings should therefore be used with caution because darktable does not allow separate pipelines for display preview and file output.

The target black luminance parameter sets the ground-level black of the target medium. By default it is set to the minimum non-zero value that can be encoded by the available number of bits in the output color space. Reducing it to zero means that some non-zero luminances will be mapped to 0 in the output, potentially losing some detail in the very darkest parts of the shadows. Increasing this slider will produce raised, faded blacks that can provide something of a “retro” look.

### target middle-gray

This is the middle-gray of the output medium that is used as a target for the S-curve’s central node. On gamma-corrected media, the actual gray is computed with the gamma correction ( $\text{middle-gray}^{(1/\text{gamma})}$ ), so a middle-gray parameter of 18% with a gamma of 2.2 gives an actual middle-gray target of 45.87%.

### target white luminance

This parameter allows you to set the ceiling level white of the target medium. Set it lower than 100% if you want dampened, muted whites to achieve a retro look.

To avoid double-ups and washed-out images, *filmic rgb* applies a “gamma” compression reverting the output ICC gamma correction, so the middle-gray is correctly remapped at the end. To remove this compression, set the destination power factor to 1.0 and the middle-gray destination to 45%.

## options

### color science

This setting defaults to v5 (2021) for new images, and defines the algorithms used by the *filmic rgb* module (such as the extreme luminance desaturation strategy). To revert to the behaviour of previous versions of *filmic rgb*, set this parameter to v3 (2019) or v4 (2020). If you have previously made edits to an image using older versions of *filmic rgb*, this will already be set to one of these versions in order to provide backward compatibility for those edits.

## preserve chrominance

Define how the chrominance should be handled by *filmic rgb* – either not at all, or using one of the three provided norms.

When applying the S-curve transformation independently on each color, the proportions of the colors are modified, which modifies the properties of the underlying spectrum, and ultimately the chrominance of the image. This is what happens if you choose “no” in the preserve chrominance parameter. This value may yield seemingly “better” results than the other values, but it may negatively impact later parts of the pipeline, for example, when it comes to global saturation.

The other values of this parameter all work in a similar way. Instead of applying the S-curve to the R, G and B channels independently, *filmic rgb*, divides all the three components by a norm (N), and applies the S-curve to N. This way, the relationship between the channels is preserved.

The value of the preserve chrominance parameter indicates which norm is used (the value used for N):

- *no* means that the ratios between the RGB channels are not preserved. This will tend to saturate the shadows and desaturate the highlights, and can be helpful when there are out-of-gamut blues or reds.
- *max RGB* is the maximum value of the R, G and B channels. This is the same behaviour as the original version of the *filmic rgb* module. It tends to darken the blues, especially skies, and to yield halos or fringes, especially if some channels are clipped. It can also flatten the local contrast somewhat.
- *luminance Y* is a linear combination of the R, G and B channels. It tends to darken and increase local contrast in the reds, and tends not to behave so well with saturated and out-of-gamut blues.
- *RGB power norm* is the sum of the cubes of the R, G, and B channels, divided by the sum of their squares ( $R^3 + G^3 + B^3$ )/( $R^2 + G^2 + B^2$ ). It is usually a good compromise between the max RGB and the luminance Y values.
- *RGB euclidean norm* has the property of being RGB-space-agnostic, so it will yield the same results regardless of which working color profile is used. It weighs more heavily on highlights than the power norm and gives more highlights desaturation, and is probably the closest to a color film look.

There is no “right” choice for the norm, and the appropriate choice depends strongly on the image to which it is applied. You are advised to experiment and decide for yourself which setting gives the most pleasing result with the fewest artifacts.

## contrast in highlights

This control selects the desired curvature at the highlights end of the *filmic rgb* spline curve. The default setting (*safe*) is guaranteed not to over- or under-shoot but has quite muted contrast near white. Selecting *hard* places a tighter constraint on the slope of the spline, which makes the curve sharper and hence introduces more tonal compression in the highlights. Selecting *soft* loosens this constraint, resulting in a gentler curve with less tonal compression in the highlights.

## contrast in shadows

This control selects the desired curvature at the shadows end of the *filmic rgb* spline curve. The default setting (*safe*) is guaranteed not to over- or under-shoot but has quite muted contrast near black. Selecting *hard* places a tighter constraint on the slope of the spline, which makes the curve sharper and hence introduces more tonal compression in the shadows. Selecting *soft* loosens this constraint, resulting in a gentler curve with less tonal compression in the shadows.

## use custom middle-gray values

Enabling this setting makes the *middle-gray luminance* slider visible on the [scene](#) tab. With the current version of *filmic rgb*, you are advised to use the *exposure* module to set the middle-gray level, so this setting is disabled by default (and the *middle-gray luminance slider* is hidden).

## auto-adjust hardness

By default, this setting is enabled, and *filmic rgb* will automatically calculate the power function (aka “gamma”) to be applied on the output transfer curve. If this setting is disabled, a *hardness* slider will appear on the [look](#) tab so that value can be manually set.

## iterations of high-quality reconstruction

Use this setting to increase the number of passes of the highlight reconstruction algorithm. More iterations means more color propagation into clipped areas from pixels in the surrounding neighbourhood. This can produce more neutral highlights, but it also costs more in terms of processing power. It can be useful in difficult cases where there are magenta highlights due to channel clipping.

The default reconstruction works on separate RGB channels and has only one iteration applied, whereas the *high quality* reconstruction uses a different algorithm that works on RGB ratios (which is a way of breaking down chromaticity from luminance) and can use several iterations to gradually propagate colors from neighbouring pixels into clipped areas. However, if too many iterations are used, the reconstruction can degenerate, which will result in far colors being improperly inpainted into clipped objects (color bleeding) – for example white clouds being inpainted with blue sky, or the sun disc shot through trees being inpainted with leaf-green.

**add noise in highlights**

This artificially introduces noise into the reconstructed highlights to prevent them from looking too smooth compared to surrounding areas that may already contain noise. This can help to blend the reconstructed areas more naturally with the surrounding non-clipped areas.

**type of noise**

This specifies the statistical distribution of the added noise. It can be helpful to match the look of the artificially generated noise with the naturally occurring noise in the surrounding areas from the camera's sensor. The *poissonian* noise is the closest to natural sensor noise but is less visually pleasing than *gaussian*, which is probably closer to film grain. Also note that most denoising modules will turn the sensor noise from poissonian to slightly gaussian, so you should pick the variant that blends better into the actual noise in your image.

## 8.2.31. framing

Generate a frame around the image.

The frame consists of a border (with a user-defined color) and a frame line within that border (with a second user-defined color). Various options are available to control the geometry and color of the frame.

### module controls

**border size**

The size of the frame as a percentage of the underlying full image.

**aspect**

The aspect ratio of the final module output (i.e. the underlying image plus the frame)

**orientation**

The orientation of the frame (portrait/landscape). Select 'auto' for darktable to choose the most reasonable orientation based on the underlying image.

**horizontal/vertical position**

Select from a set of pre-defined ratios to control where the underlying image will be positioned on the horizontal/vertical axis. You can also right click and enter your own ratio as "x/y".

**frame line size**

The percentage of the frame line size, relative to the border size at its smallest part.

**frame line offset**

The position of the frame line, relative to the underlying image. Choose 0% for a frame line that touches the image. Choose 100% for a frame line that touches the outer border.

**border color / frame line color**

A pair of color selectors which allow the border and frame line colors to be defined. Clicking on the colored field will open a color selector dialog which offers a choice of commonly-used colors, or allows you to define a color in RGB color space. You can also activate a color picker to take a color probe from the image.

**show guides**

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.32. global tonemap (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [filmic rgb](#) module instead.**

---

Compress the tonal range of an HDR image into the limited tonal range of a typical LDR output file.

Global tonemap processes each pixel of an HDR image, without taking the local surrounding into account. This is generally faster than local [tone mapping \(deprecated\)](#), but might lead to less convincing results with very high-dynamic-range scenes.



## module controls

### operator

The operator to use. *Reinhard*, *Filmic* and *Drago* global tonemap operators are available. Depending on the selected operator, different parameters can be adjusted.

Some operators are fully self-adjusting, and do not require specific controls.

### bias (*Drago* operator only)

This parameter influences the contrast of the output image. It is an essential parameter for adjusting the compression of high values and the visibility of details in dark areas. A value of 0.85 is recommended as a starting point.

### target (*Drago* operator only)

A scale factor to adjust the global image brightness to the brightness of the intended display. It is measured in cd/m, and should match the brightness of your output device. Higher values lead to a brighter image, while lower values lead to a darker image.

### detail (all operators)

This parameter controls how much detail is preserved from the original input image and transferred back into the output image after tonemapping.

## 8.2.33. graduated density

Simulate a graduated density filter in order to correct exposure and color in a progressive manner.

A line is shown on screen allowing the position and rotation of the gradient to be modified with the mouse.

This module is known to provoke banding artifacts under certain conditions. You should consider activating the [dithering](#) module to alleviate these issues.

## module controls

### density

Set the density of the filter (EV). A low value underexposes slightly whereas a high value creates a strong filter.

### hardness

The progressiveness of the gradient. A low value creates a smooth transition, whereas a high value makes the transition more abrupt.

### rotation

The rotation of the filter. Negative values rotate clockwise. The rotation can also be set by dragging the end of the gradient line with the mouse.

### hue

Choose a hue to add a color cast to the gradient.

### saturation

The saturation of the color cast to add to the gradient (defaults to a neutral color cast of 0)

## 8.2.34. grain

Simulate the grain of analog film. The grain is processed on the L channel of Lab color space.

## module controls

### coarseness

The grain size, scaled to simulate an ISO number.

### strength

The strength of the effect.

## 8.2.35. haze removal

Automatically reduce the effect of dust and haze in the atmosphere. This module may also be employed more generally to give pictures a color boost specifically in low-contrast regions of the image.

Haze absorbs light from objects in the scene but it is also a source of diffuse background light. The haze removal module first estimates, for each image region, the amount of haze in the scene. It then removes the diffuse background light according to its local strength and recovers the original object light.

Setting both of the module's controls to unity maximizes the amount of haze removal but is also likely to produce some artifacts. Removing the atmospheric light entirely may render the image flat and result in an unnatural looking style. Optimal values are typically below unity and are rather image dependent, but also a matter of personal aesthetic preferences.

### module controls

#### strength

The amount of haze removal. At unity, the module removes 100 percent of the detected haze up to the specified distance. Negative values increase the amount of haze in the image.

#### distance

Limit the distance up to which haze is removed. For small values, haze removal is restricted to the foreground of the image. Haze is removed from the entire image if the distance parameter is set to unity. If the *strength* is negative the distance control has no effect.

## 8.2.36. highlight reconstruction

Attempt to reconstruct color information for pixels that are clipped in one or more RGB channel.

If these pixels are left partially clipped it can result in unrealistic colors appearing in the image. For example pixels with both green and blue channels clipped will appear to be red without any highlight reconstruction.

Three methods of reconstruction are available:

#### clip highlights

Clamp all pixels to the white level (i.e. clip the remaining color channels). This method is most useful in cases where clipped highlights occur in naturally desaturated objects (e.g. clouds).

#### reconstruct in LCh

Analyse each pixel with at least one clipped channel and attempt to correct the clipped pixel (in LCh color space) using the values of the other (3 for Bayer or 8 for X-Trans) pixels in the affected sensor block. The reconstructed highlights will still be monochrome, but brighter and with more detail than with "clip highlights". This method works fairly well with a high-contrast base curve, which renders highlights desaturated. As with *clip highlights* this method is a good option for naturally desaturated objects.

#### reconstruct color

Use an algorithm that transfers color information from unclipped surroundings into the clipped highlights. This method works very well on areas with homogeneous colors and is especially useful on skin tones with smoothly fading highlights. Please note that this method can produce maze-like artifacts on highlights behind high-contrast edges, for example well-exposed fine structures in front of an overexposed background.

For highlight reconstruction to be effective you need to also apply a negative EV correction in the [exposure](#) module

---

**Note:** When using the highlight reconstruction included with the [filmic rgb](#) module it may be useful to disable the *highlight reconstruction* module so that filmic has more information to work with.

---

### module controls

#### method

The method used to reconstruct highlights (see above).

#### clipping threshold

Manually adjust the clipping threshold against magenta highlights. You shouldn't need to adjust this.



## 8.2.37. highpass

A high pass filter.

This module is primarily intended to be used in combination with a [blend mode](#) . For example, try using the module with a blend mode of “soft light” for high pass sharpening.

---

**Note:** *This module performs blurs in Lab color space, which can result in undesirable effects, and is no longer recommended. Instead, use the [contrast equalizer](#) module for fine sharpness or the [local contrast](#) module for general sharpness.*

---

### Module Controls

#### sharpness

The sharpness of the filter

#### contrast boost

The contrast boost

## 8.2.38. hot pixels

Automatically detect and eliminate hot pixels.

Hot pixels are pixels which have failed to record a light level correctly. Detected hot pixels are replaced by an average of their neighbors.

### module controls

#### threshold

How strong a pixel’s value needs to deviate from that of its neighbors to be regarded as a hot pixel.

#### strength

The blending strength of the hot pixels with their surrounding.

#### detect by 3 neighbours

Extend the detection of hot pixels – regard a pixel as hot if a minimum of only three (instead of four) neighbor pixels deviate by more than the threshold level.

#### mark fixed pixels

Visually mark the corrected pixels on the image and display a count of hot pixels that have been fixed.

## 8.2.39. input color profile

Define how darktable will interpret the colors of the image.

This module takes the color space used by the image source (e.g. camera, scanner) and converts the pixel encodings to a standardized working color space. This means that subsequent modules in the pipeline don’t need to be concerned with the specifics of the input device, and can work with and convert to/from a common working color space.

Where an image has been captured in a raw file, the input color profile module will normally apply either a standard or enhanced color matrix specific for that camera model, which will be used to map the colors into the working profile color space. If color space information is embedded in the image, the *input color profile* module will use this information when mapping the colors to the working profile color space. The user can also explicitly specify a color space for the incoming image, and can even supply a custom ICC color profile specifically made for the input device.

As part of the mapping from the input color space to the working profile space, the colors can be confined to a certain gamut using the *gamut clipping* options, which can help to mitigate some (infrequent) color artifacts. This is also influenced by the chosen [rendering intent](#) .

Note that the final color profile that will be used when exporting the image is controlled by the [output color profile](#) module.

---

## module controls

### input profile

The profile or color matrix to apply. A number of matrices are provided along with an enhanced color matrix for some camera models. The enhanced matrices are designed to provide a look that is closer to that of the camera manufacturer.

You can also supply your own input ICC profiles and put them into `$DARKTABLE/share/darktable/color/in` or `$HOME/.config/darktable/color/in` (where `$DARKTABLE` is the darktable installation directory and `$HOME` is your home directory). One common source of ICC profiles is the software that is shipped with your camera, which often contains profiles specific to your camera model. You may need to activate the [unbreak input profile](#) module to use your own profiles.

If your input image is a low dynamic range file like JPEG, or a raw file in DNG format, it might already contain an embedded ICC profile, which darktable will use by default. You can restore this default by selecting “embedded icc profile”.

### working profile

The working profile used by darktable’s processing modules. Each module can specify an alternative space that it will work in, and this will trigger a conversion. By default darktable will use “linear Rec2020 RGB”, which is a good choice in most cases.

### gamut clipping

Activate a color clipping mechanism. In most cases you can leave this control in its default “off” state. However, if your image shows some specific features such as highly saturated blue light sources, gamut clipping might be useful to avoid black pixel artifacts. See [possible color artifacts](#) for more information.

Choose from a list of RGB profiles. Input colors with a saturation that exceeds the permissible range of the selected profile are automatically clipped to a maximum value. “linear Rec2020 RGB” and “Adobe RGB (compatible)” allow for a broader range of unclipped colors, while “sRGB” and “linear Rec709 RGB” produce a tighter clipping. Select the profile that prevents artifacts while still maintaining the highest color dynamics.

## 8.2.40. invert (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [negadoctor](#) module instead.**

---

Invert scanned negatives.

## module controls

### color of film material

Clicking on the colored field will open a color selector dialog which offers you a choice of commonly used colors, or allows you to define an RGB color. You can also activate the color picker to take a color probe from your image – preferably from the unexposed border of your negative.

## 8.2.41. lens correction

Automatically correct for (and simulate) lens distortion, transversal chromatic aberrations (TCA) and vignetting.

This module identifies the camera/lens combination from the image's Exif data and uses the external [lensfun library](#) to provide correction parameters.

If your system's lensfun library has no correction profile for the automatically identified camera/lens combination, the controls for the three photometric parameters (below) are replaced with a warning message. You may try to find the right profile yourself by searching for it in the menu.

If your lens is present in the list but has not been correctly identified, this may require some adjustment within the exiv2 program (see [this post](#) for details). Note that you may need to re-import the images once such adjustments have been made as the lens name is retrieved as part of the import process.

If you can't find your lens, check if it is in the list of [currently supported lenses](#), and try running the [lensfun-update-data](#) tool. If there is still no matching profile for your lens, a [lens calibration service](#) is offered by Torsten Bronger, one of darktable's users. Alternatively you may visit the [lensfun project](#) to learn how to generate your own set of correction parameters. Don't forget to share your profile with the lensfun team!

### module controls

#### camera

The camera make and model as determined by the image's Exif data. You can override this manually and select your camera from a hierarchical menu. Only lenses with correction profiles matching the selected camera will be shown.

#### lens

The lens make and model as determined by the image's Exif data. You can override this manually and select your lens from a hierarchical menu. This is mainly required for pure mechanical lenses, but may also be needed for off-brand / third party lenses.

#### photometric parameters (focal length, aperture, focal distance)

Lens corrections depend on certain photometric parameters that are read from the image's Exif data: focal length (for distortion, TCA, vignetting), aperture (for TCA, vignetting) and focal distance (for vignetting). Many cameras do not record focal distance in their Exif data, in which case you will need to set this manually.

You can manually override all automatically selected parameters. Either take one of the predefined values from the drop-down menu or, with the drop-down menu still open, just type in your own value.

#### corrections

Choose which corrections (distortion, TCA, vignetting) darktable should apply. Change this from its default "all", if your camera has already performed some internal corrections (e.g. vignetting), or if you plan to undertake some corrections with a separate program.

#### geometry

In addition to correcting lens flaws, this module can change the projection type of your image. Set this combobox to the desired projection type (e.g. "rectilinear", "fish-eye", "panoramic", "equirectangular", "orthographic", "stereographic", "equisolid angle", "thoby fish-eye"). To correct the aspect ratio of an anamorphic lens, use the [rotate and perspective](#) module.

#### scale

Adjust the scaling factor of your image to avoid black corners. Press the auto scale button (to the right of the slider) for darktable to automatically find the best fit.

#### mode

The default behavior of this module is to *correct* lens flaws. Switch this combobox to "distort" in order to instead *simulate* the flaws/distortions of a specific lens (inverted effect).

#### TCA overwrite

Check this box to override the automatic correction parameters for TCA. This will expose the TCA red and TCA blue parameters below. Un-check the box to revert back to automatic corrections.

#### TCA red; TCA blue

Override the correction parameters for TCA. You can also use these sliders to manually set the parameters if the lens profile does not include TCA correction. Look out for colored seams at features with high contrast edges and adjust the TCA parameters to minimize those seams.

---

**corrections done**

Occasionally, for a given camera/lens combination, only some of the possible corrections are supported by lensfun's profiles. This message box will tell you which corrections have actually been applied to the image.

---

**Note:** TCA corrections will not be applied to images that have been identified as monochrome (see [developing monochrome images](#) for more information).

**Note:** The lens correction module will fill in missing data at the borders by repeating the borders' pixels. For strong corrections, this filling can be visible (especially on noisy images). Crop the image if necessary.

---

**show guides**

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.42. levels

Adjust black, white and mid-gray points.

The levels tool offers two modes of operation:

**manual**

The levels tool shows a histogram of the image, and displays three bars with handles. Drag the handles to modify the black, middle-gray and white points in absolute values of image lightness (the L value from Lab).

Moving the black and white bars to match the left and right borders of the histogram will make the output image span the full available tonal range, increasing the image's contrast.

Moving the middle bar will modify the mid-tones. Move it to the left to make the image look brighter and move it to the right to make it darker. This is often referred to as changing the image's gamma.

Three color pickers are available for sampling the black, white and gray points from the image. The "auto" button auto-adjusts the black and white point and puts the gray point exactly in the mean between them.

**automatic**

The module automatically analyses the histogram of the image, detects the left and right histogram borders, and lets you define the black point, the gray point and the white point in terms of [percentiles](#) relative to these borders.

---

**Note:** Under certain conditions, especially with highly saturated blue light sources, the *levels* module may produce black pixel artifacts. See the "gamut clipping" option of the [input color profile](#) module for information about how to mitigate this issue.

---

## module controls

**mode**

The mode of operation (automatic or manual).

**black (automatic mode only)**

The black point in percentiles relative to the left border of the histogram.

**gray**

The gray point in percentiles relative to the left and right borders of the histogram after having applied the black point and white point corrections.

**white**

The white point in percentiles relative to the right border of the histogram.

## 8.2.43. liquify

Move pixels around by applying freestyle distortions to parts of the image using points, lines and curves.

---

## nodes

Each of liquify's tools is based on nodes. A point consists of a single node and a line or curve consists of a sequence of linked nodes defining a path.

Each instance of the liquify module is limited to a maximum of 100 nodes – for more nodes, use additional instances. However, please note that the liquify module consumes a lot of system resources.

Drag the central point of a node to move the node around. The radius describes the area of the effect (distortion occurs only within this radius). To change the radius drag the handle at the circumference. A strength vector starting from the center describes the direction of the distortion, and its strength is depicted by the length of the vector. Change the vector by dragging its arrow head.

## points

A point consists of a single node and strength vector.

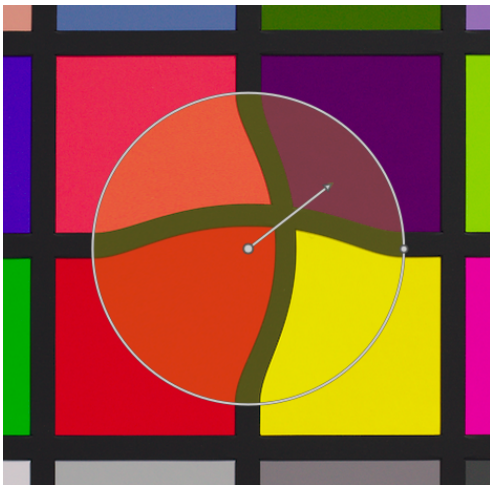
Click the point icon to activate the point tool and then click on the image to place it. Hold Ctrl while clicking on the point icon to add multiple points without having to click the icon again each time. Right-click to exit creation mode.

## point modes

The strength vector of a point has three different modes. These can be toggled by holding Ctrl and clicking on the arrowhead of the strength vector.

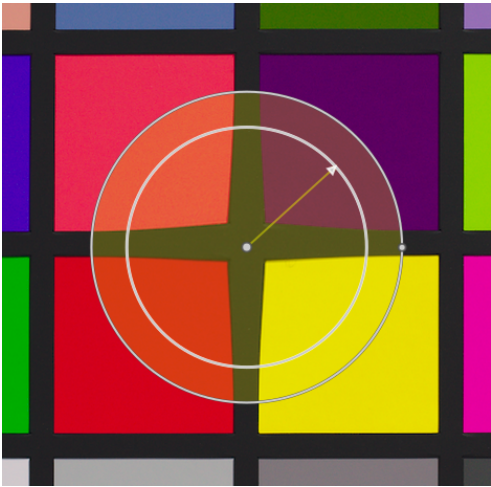
### linear

A linear distortion inside the circle, starting from the opposite side of the strength vector and following the vector's direction. This is the default mode.



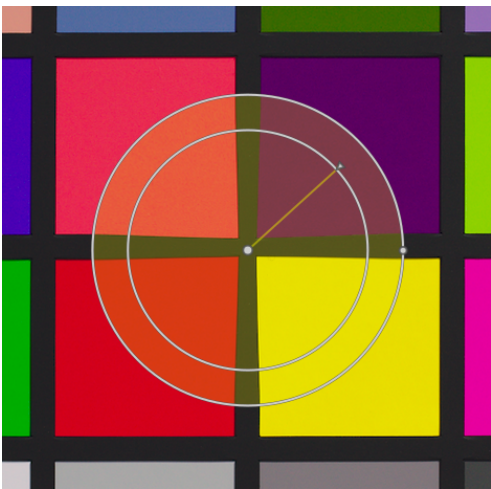
### radial growing

The strength vector's effect is radial, starting with a strength of 0% in the center and increasing away from the center. This mode is depicted by an additional circle with the arrow pointing outwards.



### radial shrinking

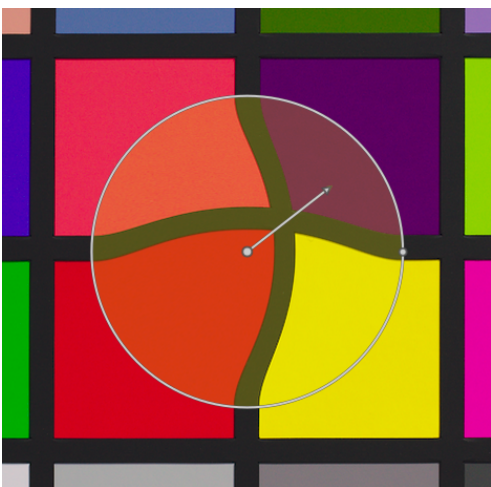
The strength vector's effect is radial, starting with a strength of 100% in the center and decreasing away from the center. This mode is depicted by an additional circle with the arrow pointing inwards.



### feathering

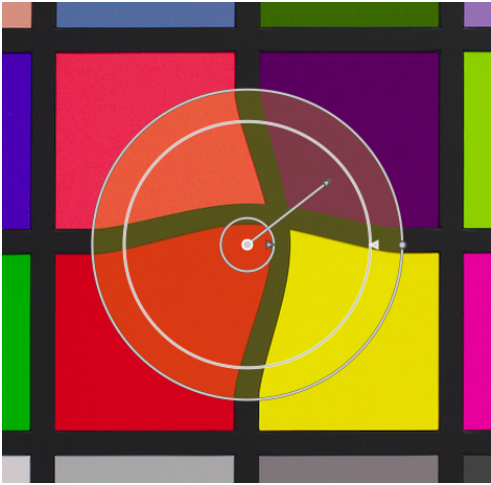
#### default mode

By default the strength varies linearly from 0% to 100% between the center and the radius of the control point. It is possible to modify the feathering effect by clicking on the center of the circle.



### feathered mode

In “feathered” mode, two control circles are displayed, which can be modified independently to feather the strength of the effect. Note that clicking the center of the circle again hides the feathering controls but does *not* return to the default mode.



### removing points

A point can be removed by right-clicking on the center of the node.

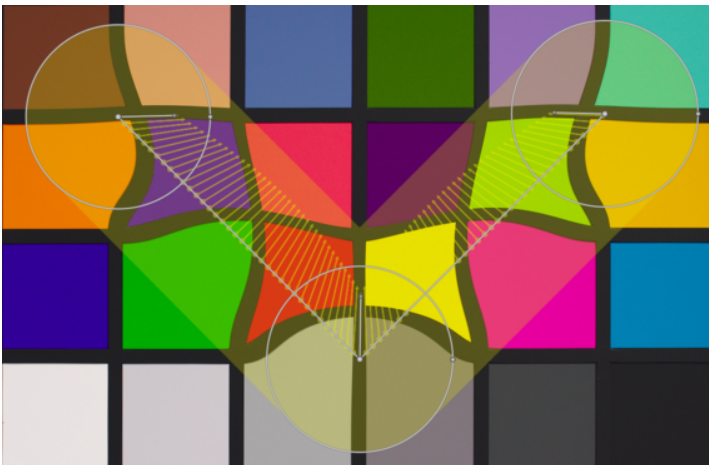
### lines and curves

Lines and curves are sequences of points linked together by straight or curved lines. The effect is interpolated by a set of associated strength vectors.

Click the appropriate icon to activate the line or curve tool and then click on the image to place a sequence of points forming the path. Right-click anywhere when the last point has been placed in order to finish drawing the line/curve.

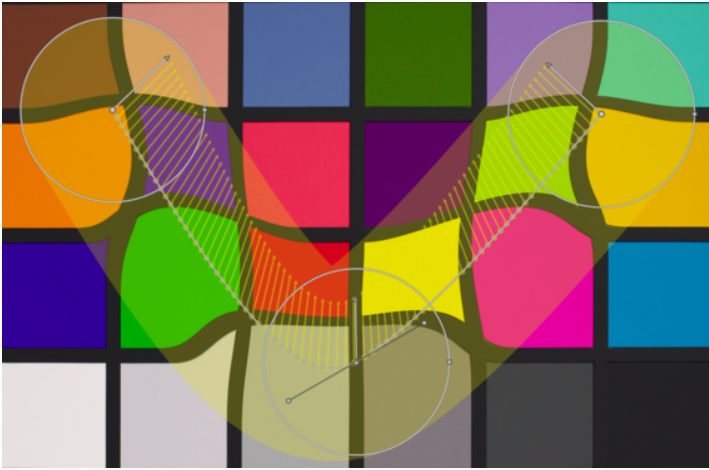
Hold Ctrl while clicking on the line/curve icon to add multiple lines/curves without having to click the icon again each time. Right-click a second time to exit creation mode after the final line or curve has been completed.

#### lines



#### curves





Ctrl+click on a line or curve segment to add a new control point. Ctrl+right-click on the center of a node to remove a control point.

Right-click on a segment to remove the shape completely. Ctrl+Alt+click on a segment to change that segment from a line to a curve and vice versa.

## link modes

Ctrl+click on the center of a node to change the way the points of a curve are linked together. There are four modes, which correspond to different ways of handling the steepness of the bezier curve using control handles:

### autosmooth

This is the default mode, in which control handles are not displayed – controls are automatically computed to give a smooth curve.

### cusp

Control handles can be moved independently. This mode is depicted by a triangle symbol in the node's center.

### smooth

Control handles always give a smooth curve. This mode is depicted by a diamond symbol in the node's center.

### symmetrical

Control handles are always moved together. This mode is depicted by a square symbol in the node's center.

## view and edit nodes

Click the node tool icon to activate or deactivate the node edit tool. This displays all currently-defined distortion objects and their controls. Alternatively you can right-click on the image at any time for the same effect.

## warps and nodes count

This information field displays the number of warps (individual distortion objects) and nodes currently in use.

## show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.44. local contrast

Enhance the image's local contrast.

This is achieved using either a *local laplacian* (default) or *unnormalized bilateral* filter. Both modes work exclusively on the L channel from Lab. The *local laplacian* filter has been designed to be robust against unwanted halo effects and gradient reversals along edges.

## module controls

### mode

Choice of *local laplacian* filter or *bilateral grid*. The following sections define the controls available for each of these modes.

## bilateral grid

### coarseness

Adjust the coarseness of the details to be adjusted.

### contrast

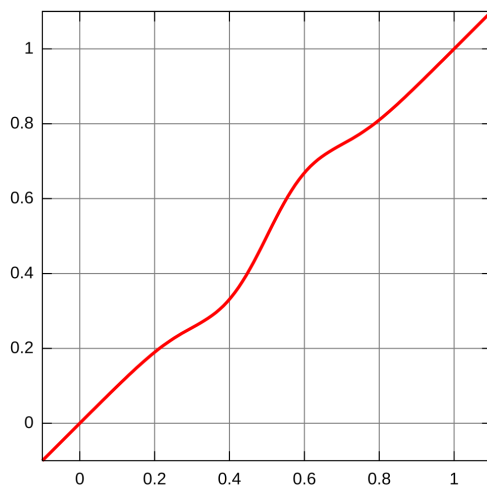
Control how strongly the algorithm distinguishes between brightness levels. Increase this parameter for a more contrasty look.

### detail

Add or remove detail. Higher values increase local contrast.

## local laplacian

To understand the parameters of the local laplacian filter, one can think of it as applying a curve to the image, similar to the following graph:



This curve is applied to the image in a way that works locally and avoids halo artifacts.

The local laplacian mode also supports shadow lifting and highlight compression, similar to the [shadows and highlights](#) module.

### detail

Add or remove detail. Higher values increase local contrast. This inserts an S shaped element in the center of the curve, to increase or decrease local contrast.

### highlights

This affects one end of the S shaped contrast curve, effectively increasing or compressing contrast in the highlights. A low value pulls the highlights down.

### shadows

Similar to the highlights parameter, this affects the other end of the contrast curve, and increases or decreases contrast in the shadows. A higher value gives more contrast in the shadows. A lower value lifts the shadows and can effectively simulate a fill light. Note that this is done with local manipulation of the image. However, this means that a completely dark image cannot be brightened in this way - only dark objects in front of bright objects are affected.

### mid-tone range

This controls the extent of the S shaped part of the contrast curve. A larger value makes the S wider, and thus classifies more values as mid-tones and fewer values as highlights and shadows. In higher dynamic range settings it can be useful to reduce this value to achieve stronger range compression, by lowering the contrast in the highlights and the shadows. Note however, that for really strong HDR scenarios this may work best in combination with a [base curve](#) that pre-compresses the range, perhaps with an approximately logarithmic curve. The exposure fusion feature in the [base curve](#) module may lead to more pleasing results at times, but is also more prone to halo effects.

This setting can cause banding artifacts in the image if pushed to extreme values. This is due to the way in which darktable computes the fast approximation of the local laplacian filter.

## 8.2.45. lowlight vision

Simulate human lowlight vision. This module can also be used to perform a day-to-night conversion.

The idea is to calculate a [scotopic vision](#) image, which is perceived by the rods rather than the cones in the eyes under low light. Scotopic lightness is then mixed with photopic value (regular color image pixel) using some blending function. This module can also simulate the [Purkinje effect](#) by adding some blueness to the dark parts of the image.

This module comes with several presets, which can be used to get a better idea of how the module works.

## module controls

### curve

The horizontal axis represents pixel lightness from dark (left) to bright (right). The vertical axis represents the kind of vision from night vision (bottom) to day vision (top).

### blue

Set the blue hint in shadows (Purkinje effect).

## 8.2.46. lowpass

Apply a low pass filter (for example, a Gaussian blur) to the image, while controlling the output contrast and saturation.

This module is primarily intended to be used in combination with a [blend mode](#). For example, try using the *local contrast mask* preset with the *overlay* blend mode.

---

**Note:** This module performs blurs in Lab color space, which can produce undesirable effects, and is no longer recommended. Instead, use the [contrast equalizer](#) module for blurring or the [tone equalizer](#) module for dynamic range compression.

---

## module controls

### radius

The radius of the blur.

### soften with

The blur algorithm to use:

- *gaussian blur*: Blur all image channels (L, a, b)
- *bilateral filter*: Blur the L channel only, while preserving edges

### contrast

Higher absolute values increase contrast. Lower absolute values reduce contrast. Negative values result in an inverted negative image. A value of zero leads to a neutral plane.

### brightness

Negative values darken the image. Positive values lighten the image.

### saturation

The color saturation. Negative values result in complementary colors by inverting the a/b channels. Higher absolute values increase color saturation. Lower absolute values reduce color saturation. A value of zero fully desaturates the image.

## 8.2.47. lut 3D

Transform RGB values with a 3D LUT file.

A 3D LUT is a tridimensional table that is used to transform a given RGB value into another RGB value. It is normally used for film simulation and color grading.

This module accepts `.cube`, `.3dl`, `.png` (haldclut) and `.gmz` files. Uncompressed 3D LUT data is not saved in the database or the XMP file, but is instead saved to the 3D LUT file path inside the 3D LUT root folder. It is therefore important to back up your 3D LUT folder properly – sharing an image with its XMP is pointless if the recipient doesn't also have the same 3D LUT file in their own 3D LUT folder.

The compressed format `.gmz` is only available when GMIC is installed. This format can hold a full library of LUTs, and LUT data loaded from this type of file *can* be saved to the database and XMP files.

---

**Note:** the module clips all values outside of the range [0,1]. You may have to reduce the range of the input before applying.

---

### usage

LUTs are most commonly used in darktable for color grading or film look simulation. For this reason, by default, the module is placed after the [filmic rgb](#) module in the pixelpipe and should be applied to a neutral image (without first applying a specific look). While you can find hundreds of free LUTs on the internet, you should note that not all of them are compatible with the darktable environment and workflow – incompatible LUTs will not produce the advertised look. To limit the risk, a color grading LUT should have been created to work with one of the available “application color spaces” (see below), for both the input and the output of the module.

Camera log LUTs (as F-log or S-Log3) are different to color-grading and film-look-simulation LUTs, and are intended to convert the camera log raw data into something (linear raw data or other color space) that darktable is able to understand. In this case the `lut 3D` module should be manually placed between the [demosaic](#) and [input color profile](#) modules. Once you have done this, you can no longer choose an “application color space”. The “input profile” of input color profile module should be aligned with the output of the LUT. *Please note that this use case has not yet been tested.*

### module controls

#### file selection

Choose the 3D LUT file to use. File selection is inactive if the 3D LUT root folder has not been defined in [preferences > processing](#).

#### application color space

A 3D LUT is defined relative to a specific color space. Choose the color space for which the selected 3D LUT file has been built. Cube files are usually related to REC.709 while most others are related to sRGB.

#### interpolation

This defines how to calculate output colors when input colors are not exactly on a node of the RGB cube described by the 3D LUT. There are three interpolation methods available: tetrahedral (default), trilinear and pyramid. Usually you won't see any difference between interpolation methods except with smaller sized LUTs.

## 8.2.48. monochrome

Quickly convert an image to black & white using a variable color filter.

To use this module, first reduce the filter size (to concentrate its effect) and move it across the hue palette to find the best filter value for your desired image rendition. Then gradually expand the filter to include more hues and achieve more natural tonality.

Although this module is easy to use, better results can usually be obtained by using the film emulation presets in the [color calibration](#) module.

---

**Note:** Under certain conditions, especially highly saturated blue light sources in the frame, this module may produce black pixel artifacts. Use the gamut clipping option of the [input color profile](#) module to mitigate this issue.

---

## module controls

### filter size/position

The default central location of the filter has a neutral effect but dragging it to an alternate position applies a filter analogous to taking a b&w photograph through a conventional color filter.

A color picker can be activated to automatically set the position and size of the filter based on the selected portion of the image. Scroll the mouse wheel to change the filter size, making the filter's range of hues more or less selective.

### highlights

Control how much to retain highlights.

## 8.2.49. negadoctor

Process scanned film negatives.

You can obtain an image of a negative using a film scanner, or by photographing it against a white light (e.g. a light table or computer monitor) or off-camera flash.

## preparation

If the image of the negative was obtained using a digital camera, then in order to obtain accurate colors in the final image, you will need to take the following points into consideration:

- When taking the photograph, adjust the exposure to fully utilise the entire dynamic range of your camera sensor – i.e. “expose-to-the-right”, so that the histogram in your camera touches the right hand side without clipping the image.
- Ensure the white balance is correctly set up to compensate for the light source used to illuminate the negative. You can take a profiling picture of the light source with no film negative in front of it, and then use the “from image area” feature in the [white balance](#) module to obtain a reference white-balance setting. This reference white balance setting can then be made into a style or simply pasted onto the images taken from your film negatives.
- Apply the standard or enhanced camera matrix in the [input color profile](#) module.

When scanning or photographing your film negative, make sure you include some unexposed part of the film within the captured image. This is required to set the *Dmin* parameter (see below). If this is not possible (e.g. your film holder completely obscures the unexposed parts of the film), you can take a separate image of an unexposed part of the film, measure the *Dmin* parameter from that image, and then paste that setting to the rest of the images from that film.

When developing the scanned/photographed film negatives, it is recommended that you disable any tone mapping modules such as [filmic rgb](#) and [base curve](#).

The *working profile* parameter in darktable's [input color profile](#) module should be set to either *linear Rec2020 RGB*, or to an ICC profile representing the actual color space of your film emulsion. Some examples of such ICC profiles may be found in the following forum posts:

- <https://discuss.pixls.us/t/any-interest-in-a-film-negative-feature-in-rt/12569/177>
- <https://discuss.pixls.us/t/input-color-profile-to-use-for-negatives/20271/13>

---

**Note:** if you want to use the [tone equalizer](#) with *negadoctor*, you'll need to move the *tone equalizer* module **after** *negadoctor* in the [pixelpipe](#), since the *tone equalizer* is not designed to work with negatives.

---

## module controls

It is strongly recommended that you set the parameters following the order in which they are presented in the GUI. Start by setting the *film stock*, then work through each of the tabs ( [film properties](#) , [corrections](#) , [print properties](#) ) in order, working from top to bottom in each tab.

When using color pickers, be careful to avoid including dust and scratches, which can skew the data taken from the sampled region.

## film stock

The first step is to choose “color” or “black and white” in the *film stock* drop-down. If you select “black and white”, any sliders that are only used for color will be hidden from view.

## film properties

This tab contains a number of basic settings. If, after adjusting these settings, your image is still not quite as you would like it, you can make further adjustments on the [corrections](#) tab. These are technical settings, and serve a similar purpose to the *scene* tab in the [filmic rgb](#) module, in that they adjust the black and white points and hence define the dynamic range of the negative.

### color of the film base

Sample an area of the base film stock from your scan. This is the area just outside of the image (an unexposed part of the film). If you are working with black and white negatives, you can leave this slider at its default value (white). If working on color film, click the color picker to the right of the color bar, which will create a bounding box covering about 98% of your image. Then, click and drag across an area of your negative which contains only unexposed film stock. This will automatically calculate values for the *D min* slider(s). It is likely at this point that your image will still look too dark, but you can correct this later.

### D min

If the *film stock* is set to “black and white”, this slider indicates the minimum value corresponding to the unexposed film stock. If the *film stock* is set to “color”, this control will consist of 3 separate sliders, one for each of the red, green and blue channels.

### D max

This slider represents the dynamic range of your film, and it effectively sets the film’s white point. Dragging this slider to the left will make the negative brighter. Dragging it to the right will make the negative darker. When adjusting this slider manually, it’s a good idea to closely watch your histogram to ensure that you don’t clip the highlights (where the histogram has been pushed over too far off the right hand side of the graph). If you click the color picker icon (on the right) negadoctor will automatically calculate this value to ensure maximal use of the histogram without clipping. To use the color picker, click and drag to draw a rectangle across only the exposed parts of the negative. Don’t include the unexposed film stock, as this will skew the result.

### scan exposure bias

This slider allows you to set the black point. It is a technical adjustment that ensures a proper zeroing of the RGB values and a spreading of the histogram between [0, 1] values for robustness in the operations that follow. Dragging this to the left will make the negative brighter. Dragging to the right will make the negative darker. When adjusting this slider manually, it’s a good idea to closely watch your histogram to ensure that you don’t clip the shadows (where the histogram is pushed too far off the left hand side of the graph). If you click the color picker negadoctor will automatically calculate any required offset. To use the color picker, select a region in the darkest lowlights, or select the entire image without including any unexposed film stock. Double-check the histogram to ensure the left part of it doesn’t clip.

## corrections

This tab contains sliders that allow you to make color cast corrections within both the shadow and highlight regions.

The settings on this tab should not be needed for most well-preserved negatives. It is mostly useful for old and poorly-preserved negatives with a decayed film base that introduces undesirable color casts. Note that the shadows color cast setting will have no effect if the *scan exposure bias* setting in the [film properties](#) tab is set to a non-zero value.

The other case where these color cast corrections may be needed is if the white balance properties of the light used to scan the film negative are significantly different to the light source under which the original film camera took the shot. For example, if you illuminate the film with an LED light, but the original shot was taken under daylight, this may require some additional color cast corrections.

### shadows color cast

These three sliders allow you to correct for color casts in the shadows by adjusting the *red*, *green* and *blue* channels individually. Use the color picker to set the sliders automatically by selecting a neutral gray shadow region requiring correction. Because the shadows sliders can also affect color casts in the highlights, it is important to finish setting the shadows sliders before moving on to the highlights sliders.

### highlights white balance

These three sliders allow you to correct the white balance in the highlights by adjusting the *red*, *green* and *blue* channels individually. Use the color picker to set the sliders automatically by selecting a neutral gray region in the highlights that is not properly balanced.

## print properties

This tab contains settings that mimic the tonal effect of the photochemical papers that would have been used to create the hard copy image if you were not developing the photo digitally. These are creative settings, and serve a similar overall purpose to the creative tone curve settings on the *look* tab of the [filmic rgb](#) module.

The *print exposure*, *paper black* and *paper grade* are analogous to the *slope*, *offset* and *power* controls in the [color balance](#) module (which in turn is based on the ASC CDL standard). These settings define a creative tone curve to enforce your contrast intent after the inversion, at the end of the module. The equation governing this slope/offset/power behaviour is:

$$RGB_{out} = ( RGB_{in} \times exposure + black )^{grade}$$

### paper black (density correction)

For this slider, select the color picker and click and drag to select a region that encompasses only the exposed part of the film negative. If you can see unexposed film stock around the edges of your image, ensure that these areas are excluded from the drawn rectangle when calculating the *paper black* setting. Paper black represents the density of the blackest silver-halide crystal available on the virtual paper. In the analog development process, this black density always results in non-zero luminance, but the digital pipeline usually expects black to be encoded with a zero RGB value. This slider setting lets you remap paper black to pipeline black via an offset. You can use the color picker to select a region of the image that should be mapped to black in the final image.

### paper grade (gamma)

This slider is your gamma (contrast) control, and it defaults to a value of 4. If all has gone well, this value (4) minus the value of D max (from the “film properties” tab) should normally leave you with a value between 2 and 3.

### paper gloss (specular highlights)

This slider is essentially a highlights compression tool. As you drag this slider to the left, you will see in the histogram that the highlight values are being compressed (pushed to the left). Adjust this accordingly, so that your highlights are not clipped in the histogram. You can also use this to simulate a matte photo print with low-contrast highlights.

### print exposure adjustment

This slider offers one final opportunity to correct any clipping of the highlights. If you have followed all the previous instructions carefully, you shouldn't need to adjust this setting. Note that you can increase the print exposure while at the same time decreasing the paper gloss, which allows you to brighten the mid-tones without losing any highlights. You can use the color picker to select the brightest highlights, or select the entire image without including any unexposed film stock. This will set the exposure so that the brightest part of the selected region is not clipped. Double-check the histogram to make sure that the right part of the histogram doesn't clip.

## 8.2.50. orientation

Rotate the image 90 degrees at a time or flip the image horizontally and/or vertically.

The module is enabled by default and the orientation (rotation) is automatically set based on the image's Exif data.

The orientation can also be set using the [selected images](#) module in the [lighttable](#) view.

## module controls

### transform

Double click the label to reset to the default transformations



#### rotate counter-clockwise

Rotate the image 90 degrees counter-clockwise



#### rotate clockwise

Rotate the image 90 degrees clockwise



#### flip horizontally

Flip the image (mirror) horizontally



#### flip vertically

Flip the image (mirror) vertically



### show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.51. output color profile

Manage the output profile for export and the rendering intent to be used when mapping between color spaces.

darktable comes with pre-defined profiles *sRGB*, *Adobe RGB*, *XYZ* and *linear RGB*. You can provide additional profiles by placing them in `$DARKTABLE/share/darktable/color/out` and `$HOME/.config/darktable/color/out` (where `$DARKTABLE` is the darktable installation directory and `$HOME` is your home directory).

The output color profile may also be defined within the [export](#) module.

## module controls

### output intent

The rendering intent for output/export. Rendering intent can only be selected when using LittleCMS2 to apply the output color profile (this can be changed in [preferences > processing](#) ). If darktable's internal rendering routines are used, this option is hidden. For more details see [rendering intent](#) .

### output profile

The profile used to render colors for output/export. The profile data will be embedded into the output file (if supported by the file format) allowing other applications to correctly interpret its colors. As not all applications are aware of color profiles, the general recommendation is to stick to sRGB unless you know what you are doing and have a good reason to do otherwise.

## 8.2.52. raw black/white point

Define camera-specific black and white points.

This module is activated automatically for all raw images. Default settings are applied for all supported cameras. Changes to the defaults are not normally required.

## module controls

### black level 0-3

The camera-specific black level of the four pixels in the RGGGBayer pattern. Pixels with values lower than this level are not considered to contain valid data.

### white point

The camera-specific white level. All pixels with values above this are likely to be clipped and will be handled accordingly in the [highlight reconstruction](#) module.

## 8.2.53. raw chromatic aberrations

Correct chromatic aberrations of raw images.

As this module operates on non-demosaiced raw data, the underlying model assumes an uncropped photographic image as input. It is therefore likely to fail when you zoom in, as it is only able to operate on the visible portion of the image. The module will therefore be bypassed at high zoom levels, and a "bypassed" message will appear at the top of the module. This limitation only applies to interactive work and not to the final export.

This module currently only works for raw images recorded with a Bayer sensor (the sensor used in the majority of cameras) – for other types of image, you should use the [chromatic aberrations](#) module instead.

The module will also not apply any corrections to any photos that have been identified as monochrome (see [developing monochrome images](#) for more information).

## module controls

### iterations

The number of iterations. For most images, "twice" is sufficient, and is the default value. Occasionally, increasing this control can give better results.

**avoid colorshift**

If the module causes purple tinting, tick this box to apply a correction.

## 8.2.54. raw denoise

Perform denoising on raw image data before it is [demosaiced](#) .

This module has been ported from [dcraw](#) .

### module controls

**noise threshold**

The threshold for noise detection. Higher values lead to stronger noise removal and greater loss of image detail.

**coarse/fine curves**

The noise of an image is usually a combination of fine-grained and coarse-grained noise. These curves allow the image to be denoised more or less depending on the coarseness of the visible noise. The left of the curve will act on very coarse grain noise, while the right of the curve will act on very fine grain noise.

Raising the curve will result in more smoothing, while lowering it will result in less smoothing. As an example, you can preserve very fine-grained noise by pulling down the rightmost point of the curve to the minimum value.

If you are tackling chroma noise with a [blend mode](#) , you can raise the rightmost part of the curve quite high, as colors do not change a lot on fine grain scales. This will help especially if you see some isolated pixel left un-denoised.

The best way to use the R, G, and B curves is to examine each of the channels in turn using the [color calibration](#) module in gray mode, denoise that channel, and then repeat for the other channels. This way, you can take into account the fact that some channels may be noisier than others. Beware that guessing which channel is noisy without actually seeing the channels individually is not straightforward and can be counterintuitive. A pixel which is completely red may not be caused by noise on the R channel, but actually by noise on G and B channels.

See the [wavelet](#) section for more details.

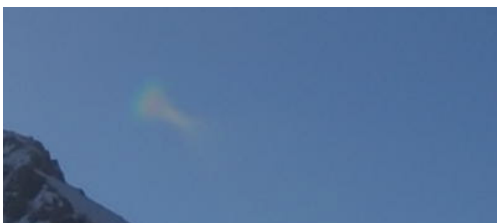
## 8.2.55. retouch


Remove unwanted elements from your image by cloning, healing, blurring and filling using drawn shapes.

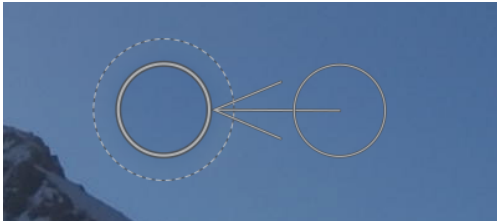
This module extends the capabilities of the deprecated [spot removal](#) module (equivalent to this module's "clone" tool) by including a "heal" tool (based on the heal tool from GIMP), as well as "fill" and "blur" modes. It can also take advantage of [wavelet decomposition](#) , allowing the image to be separated into layers of varying detail (from coarse to fine) which can be selectively retouched before being recombined to produce the output image.

### clone and heal

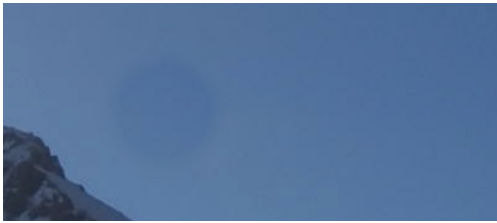
Cloning allows a part of the image (the *target*) to be hidden by replacing it with an area copied from elsewhere in the image (the *source*). For example, you may wish to remove a small cloud from a blue sky:




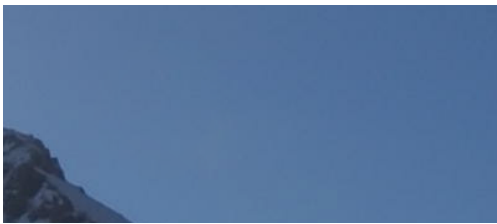
The simplest way to do this is with the *cloning tool*  . The following example uses a circular shape to clone out the cloud using the circle of blue sky beside it:



In many cases, the edges of the *source* shape won't precisely match the surroundings of the *target*, leading to unnatural looking results. In this example, the sample of sky we chose as the source was slightly darker than the target, leaving a faint outline of the circular shape used in the cloning process:



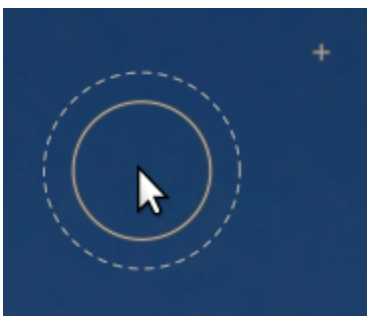
In such cases, the *heal* tool  is more appropriate. With this tool, the color and luma of the sample is blended to fit better with the surroundings. In this example, using *heal* instead of *clone* produces a much more pleasing result:



## source and target shapes

Once you have chosen *heal* or *clone* mode, you must choose the shape you wish to use (*circle*, *ellipse*, *path* or *brush* - see the [drawn masks](#) section for details). The source and target patches will both use the same shape.

When you hover over the image with your mouse, a plus symbol (+) will appear to indicate where the *source* shape will be placed by default. Your normal mouse cursor will indicate the position of the *target* shape:



It is recommended that you choose the position of the source shape first, followed by the position of the target, as follows:

- Shift+click to set the position of the source shape in “relative mode”. The position of the “plus” (+) symbol will move to the clicked location and will remain in a fixed position *relative to your cursor* until you click on the image to begin placing your target shape. If you place subsequent shapes without first changing the target location, the source shape will be placed at the same offset from the target shape as was used for the first stroke.
- Ctrl+Shift+click to change the position of the source shape in “absolute” mode. As above, the position of the “plus” (+) symbol will move to the clicked location, but will *remain in the same absolute position* even if you move your mouse. You can then click on the image to begin placing the target shape. If you place subsequent shapes without first changing the target location, exactly the same source position will be used, fixed in the absolute coordinate system of the image.

Once you have placed your source and target shapes on the image, they can be adjusted manually with your mouse.

**Note:** For *circle* and *ellipse* shapes only, you can place both the source and target shapes with a single click-and-drag motion: Click on the desired target location and then drag, releasing your mouse button when you reach the desired source location. This operation does not affect the placement of subsequent shapes.

## fill and blur

The *clone* and *heal* tools both require the use of another part of the image to “fill in” the region being hidden. Sometimes there is no suitable sample in the image to use for this purpose. In such cases, the *retouch* module offers two further options:



### fill tool

Fill the drawn region with a selected color.



### blur tool

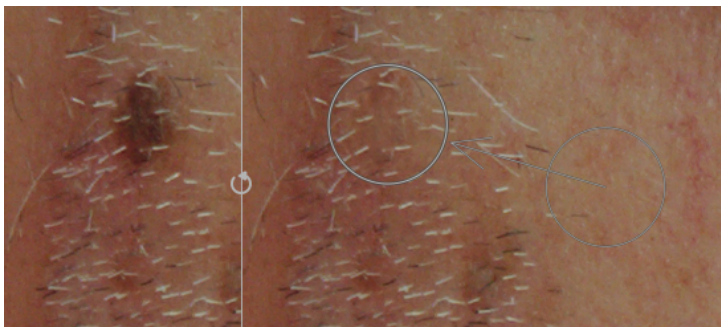
Apply a blur to the drawn region, smoothing out any details.

These two options are most useful when used together with wavelet decomposition, where they can be used to smooth over features within a selected detail layer.

## wavelet decomposition

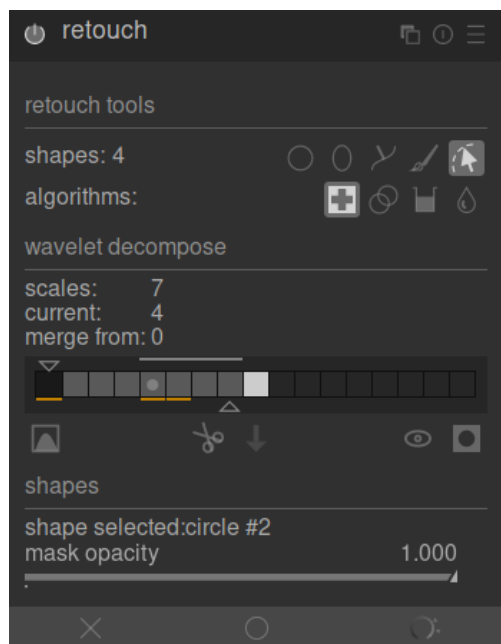
Wavelets allow an image to be decomposed into a number of layers each containing varying levels of detail, so that you can work on each detail layer independently and then recombine them at the end. This is particularly useful in portrait photography, where you can deal with skin blotches and blemishes on a coarse layer of detail, leaving the finer skin texture untouched. The [wavelets](#) section provides more information on the decomposition process.

This method can be used with the healing tool, for example, to paint over a spot that appears in one of the coarse detail layers, while leaving the hairs in the fine detail layers intact:



It can also be used with the blur tool to even out coarse blotches in the skin, again without impacting the finer details (see the [wavelets](#) section for details on this technique).

## module controls



### retouch tools


The *retouch tools* section consists of two items:

#### shapes

The number after the *shapes* label indicates how many shapes have been placed on the image, either directly or within a wavelet layer.

Click on one of the shape icons to draw a new shape on the image (see [drawn masks](#) for details).

Ctrl+click on a shape icon to draw multiple shapes continuously (right-click to cancel).

Click the *show and edit shapes*  button to show and edit any existing shapes for the currently-selected wavelet scale.

#### algorithms

Choose a retouching algorithm (clone, heal, fill or blur). Ctrl+click to change the algorithm used for the currently-selected shape. Shift+click to set the default algorithm (used for new images or when you reset module parameters).

### wavelet decompose

The *wavelet decompose* section centres around a bar graph that shows how the image has been decomposed into detail (scale) layers. The key features of the bar graph are:

- The black square on the left represents the entire image before decomposition.
- The gray squares show the various wavelet detail layers, with fine details to the left, and coarse details to the right.
- The white square on the far right represents the residual image (the remainder of the image after the detail layers have been extracted).
- A light gray dot in a square indicates the currently-selected layer. Click on another square to select a different layer.
- The light gray bar running along the top indicates which levels of detail are visible at the current zoom level. Zoom in to see finer details.
- The triangle at the bottom shows how many layers the image has been decomposed into. Drag the triangle to the right to create more layers. Drag it to the left to decrease the number of layers. By default no wavelet decomposition is performed.
- The triangle at the top shows the current value of the “merge from” parameter (see below).
- The orange lines under the squares indicate which layers have retouch edits applied.

The remaining items in this section are:

**scales**

Shows how many detail layers the image has been decomposed into. Zero indicates that no wavelet decomposition has been performed.

**current**

Shows which layer is currently selected (also marked with the light gray dot on the bar graph).

**merge from**

This setting allows a single edit to be applied to multiple consecutive scales within a group starting from the coarsest scale (excluding the residual image) down to the scale selected. For example, if “merge from” is set to 3 and the maximum scale is 5 then all edits that are added to scale 5 will be applied to scales 3, 4 and 5. Edits added to scale 4 will be applied to scales 3 and 4, and edits added to scale 3 will be applied to scale 3 only. If you set *merge from* to 0, then merging is disabled, and all edits apply only to the scale that they are defined in. Setting *merge from* to the highest scale available (in this example, 5) also disables merging.

```


merge_from
  v
0  1  2  3  4  5  scale
      <-----0  scale 5 edits
      <---0     scale 4 edits
      0         scale 3 edits
    0          scale 2 edits
  0           scale 1 edits

```

**display wavelet scale**

Display the currently-selected wavelet layer on the center image. Selecting this option brings up an additional control – *preview single scale*.

**preview single scale**

An additional control that allows the black, white and gray points of the wavelet scale preview to be adjusted to make it easier to see. Click the  to set these values automatically. This does not affect the module’s operation – only the wavelet scale preview.

**cut**

Cut all shapes from the currently-selected layer and place them into the clipboard.

**paste**

Move the shapes on the clipboard to the currently-selected layer.

**temporarily switch off shapes**

Toggle all shapes (whether on the current layer or not) on or off, temporarily removing the module’s effect.

**display masks**

Show the target shapes associated with the currently-selected layer with a yellow overlay.

## shapes

This section allows you to modify settings related to the currently-selected shape:

**shape selected**

Indicates which shape is currently selected, and what type of shape it is.

**fill mode**

If the *fill* algorithm has been chosen for the currently-selected shape, choose whether to “erase” or fill the selected shape with a chosen “color”.

**fill color**

If a *fill mode* of “color” has been chosen, select the color to fill the shape with. You can click to select or enter a custom rgb value or use the color picker to take a sample from the image.

**brightness**

If the *fill* algorithm has been chosen for the currently-selected shape, fine-tune the color’s brightness. This slider also works in “erase” mode.

**blur type**

If the *blur* algorithm has been chosen for the currently-selected shape, choose whether to apply a “gaussian” or “bilateral” blur.

**blur radius**

If the *blur* algorithm has been chosen for the currently-selected shape, choose the radius of the blur.

**mask opacity**

Alter the opacity of the mask associated with the currently-selected shape. An opacity of 1.0 indicates that the shape is completely opaque and the module’s effect is fully applied, whereas a value less than 1.0 indicates that the effect applied by the shape is blended with the underlying image to the degree indicated by the slider.

## show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## 8.2.56. rgb curve

A classic digital photography tool to alter an image’s tones using curves.

This module is very similar to the [tone curve](#) module but works in RGB color space.

Activate the color picker on the left to show the picked values in the graph (Ctrl+click or right-click to use the picker in area mode). Numerical (Lab) values of the input and output (see below) at the selected spot or area are shown at the top left of the widget.

A second color picker to the right can be used to automatically create new nodes based on the sampled area. Ctrl+click+drag to alter the created nodes to have a positive curve for the selected area; Shift+click+drag to create a negative curve.

## module controls

Please refer to the [curves](#) section for more details about how to modify curves including the **interpolation method** and **preserve colors** controls.

**mode**

RGB is a linear color space designed to capture and display images in additive synthesis. It is related to the capture and display media and does not isolate color and lightness information in the same way that Lab and XYZ color spaces do. This module works in ProPhoto RGB. Adding contrast in RGB space is known to desaturate highlights and boost saturation in the shadows, but this has been proven to be the most reliable way to edit contrast, and is the standard method in most software

Depending on the desired intent you can apply the RGB curve in two different modes

- *RGB, linked channels*: Apply the L-channel curve to all three channels in the RGB color space.
- *RGB, independent channels*: R, G and B curves can be adjusted independently.

**compensate middle gray**

Select this to change the histogram display in the module. This option does not alter the processing but may assist when editing the curve.



## 8.2.57. rgb levels

Adjust black, white and mid-gray points in RGB color space. This module is similar to the [levels](#) module, which works in Lab color space.

The rgb levels tool shows a histogram of the image, and displays three bars with handles. Drag the handles to modify the black, middle-gray and white points in lightness (in “RGB, linked channels” mode) or independently for each of the R, G and B channels (in “RGB, independent channels” mode).

Moving the black and white bars to match the left and right borders of the histogram will make the output image span the full available tonal range. This will increase the image’s contrast.

Moving the middle bar will modify the mid-tones. Move it to the left to make the image look brighter and move it to the right to make it darker. This is often referred to as changing the image’s gamma.

Three color pickers are available for sampling the black, white and gray points from the image.

---

**Note:** Under certain conditions, especially with highly saturated blue light sources, the *levels* module may produce black pixel artifacts. See the “gamut clipping” option of the [input color profile](#) module for information about how to mitigate this issue.

---

## module controls

### mode

The mode of operation. “RGB, linked channels” (default) provides a single levels tool which updates all channels, taking into account the selected color preservation method (see “preserve colors” below). “RGB, independent channels” provides separate levels controls for each of the R, G and B channels.

### auto

Auto-adjust the black and white point and put the gray point exactly in the mean between them. Use the color picker to auto-adjust based on a selected region of the image.

### preserve colors

Choose a color preservation method when using “RGB, linked channels” mode (default “luminance”).

## 8.2.58. rotate and perspective

Automatically correct for converging lines, a form of perspective distortion. The underlying mechanism is inspired by Markus Hebel’s [ShiftN](#) program. This module also allows for the rotation of the image to be adjusted.

Perspective distortions are a natural effect when projecting a three dimensional scene onto a two dimensional plane and cause objects close to the viewer to appear larger than objects further away. Converging lines are a special case of perspective distortions frequently seen in architectural photographs – parallel lines, when photographed at an angle, are transformed into converging lines that meet at some vantage point within or outside of the image frame.

This module is able to correct converging lines by warping the image in such a way that the lines in question become parallel to the image frame. Corrections can be applied in a vertical and horizontal direction, either separately or in combination. In order to perform automatic correction the module is able to analyze the image for suitable structural features consisting of line segments. You can also set the line structures manually by drawing a “perspective rectangle” or drawing multiple horizontal and vertical lines onto the image. Based on these (automatic or manually-drawn) line segments a fitting procedure is initiated, which determines the best values for the module’s parameters.

As the most common use case for this module is for rotation, the perspective correction controls are hidden by default. Click the “perspective” header to expand the controls.


While the module is active (and none of the structure buttons are selected) you can right-click and drag anywhere on the image to define a horizontal or vertical line. This will cause the rotation parameter to be automatically adjusted to make the drawn line horizontal/vertical with respect to the image frame.

## perspective correction workflow


### structure

The first step is to obtain details about the horizontal and/or vertical structures in the image. Three alternative methods are provided to do this:

#### manually draw structure lines


Click on the  icon to enable line drawing mode and then click-and-drag on the image to draw lines that you wish to become horizontal or vertical. The module will automatically detect whether the lines are horizontal or vertical and color them green or blue, respectively. Draw as many lines as you wish (the more lines, the better the fit mechanism will work) and then click on one of the “fit” icons to complete the process. You can re-enter this mode to edit your drawn lines at any time. Edit a line by clicking and dragging on the line or the end nodes, and right-click a line to delete it. Once you are happy with your changes, re-select a “fit” icon to complete the process.

#### manually define a perspective rectangle

Click on the  icon to enable perspective rectangle drawing mode. This will draw a rectangle on the screen and you can grab and move the corners of the rectangle so that the left and right sides fall on lines you wish to make vertical, and the top and bottom fall on lines you wish to make horizontal. Once you are happy with your rectangle, click one of the “fit” icons to complete the process. You can re-enter this mode to edit your drawn rectangle at any time. Once you are happy with your changes, re-select a “fit” icon to complete the process.

This method is similar to how “keystone” correction works in the [crop and rotate](#) module.

#### automatically detect structure

Click the  icon to analyze the image for structural elements – darktable will automatically detect and evaluate line elements. Shift+click to apply a contrast enhancement step before performing further analysis. Ctrl+click to apply an edge enhancement step before performing further analysis. Both variations can be used alone or in combination if the default analysis is not able to detect a sufficient number of lines.

Only lines that form a set of vertical or horizontal converging lines are used for subsequent processing steps. Line segments are displayed as overlays on the image canvas, with the type of line identified by color as follows:

##### **green**

Vertical converging lines

##### **red**

Vertical lines that do not converge

##### **blue**

Horizontal converging lines

##### **yellow**

Horizontal lines that do not converge

##### **gray**

Other lines that are not of interest to this module




Lines marked in red or yellow are regarded as outliers and are not taken into account during the automatic fitting step. This outlier elimination involves a statistical process using random sampling which means that each time you press the “get structure” button the color pattern of the lines will look slightly different.

You can manually change the status of line segments: Left-Click on a line to select it (turn the color to green or blue) and Right-click to deselect it (turn the color to red or yellow). If you keep the mouse button pressed, you can use a sweeping action to select/deselect multiple lines in a row. The size of the select/deselect brush can be changed with the mouse wheel. Hold down the Shift key and keep the left or right mouse button pressed while dragging to select or deselect all lines in the chosen rectangular area.

Once you are happy with the detected lines, select a “fit” icon to complete the process.

## fit

Once you are happy with the identified horizontal and vertical lines, using one of the methods above, click on one of the “fit” icons to automatically set the module’s parameters based on the defined structure. The image and the overlaid lines are then displayed with perspective corrections applied.

You may choose to automatically apply just the vertical corrections , just the horizontal corrections , or both together . Ctrl+click on any of the icons to apply a rotation without the lens shift. Shift+click on any of the icons to apply the lens shift without any rotation.

## rotate

Once you are happy with the applied perspective corrections, you may wish to perform a final rotation correction either by adjusting the rotation parameter or right-clicking and dragging the image to define a horizontal/vertical line.

## module controls

### rotation

Control the rotation of the image around its center to correct for a skewed horizon.

### automatic cropping

When activated, this feature crops the image to remove any black areas at the edges caused by the distortion correction. You can either crop to the “largest area”, or to the largest rectangle that maintains the original aspect ratio (“original format”). In the latter case you can manually adjust the automatic cropping result by clicking in the clip region and moving it around. The size of the region is modified automatically to exclude any black areas.

### lens shift (horizontal)

Correct converging horizontal lines (i.e. to make the *blue* lines parallel).

### lens shift (vertical)

Correct converging vertical lines (i.e. to make the *green* lines parallel). In some cases you can obtain a more natural looking image if you correct vertical distortions to an 80 ~ 90% level rather than to the maximum extent. To do this, reduce the correction slider after having performed the automatic correction.

### shear

Shear the image along one of its diagonals. This is required when correcting vertical and horizontal perspective distortions simultaneously.

### guides

When activated, a grid is overlaid on the image to help you judge the quality of the correction.

### lens model

This parameter controls the lens focal length, camera crop factor and aspect ratio that used by the correction algorithm. If set to “generic” a lens focal length of 28mm on a 35mm full-frame camera is assumed. If set to “specific”, the focal length and crop factor can be set manually using the sliders provided.

### focal length

If the *lens model* is set to “specific”, set the lens focal length. The default value is taken from the image’s Exif data, and can be overridden by adjusting the slider manually.

### crop factor

If the *lens model* is set to “specific”, set the camera crop factor. You will normally need to set this value manually.

### aspect adjust

If the *lens model* is set to “specific”, this parameter allows for a free manual adjustment of the image’s aspect ratio. This is useful for “unsqueezing” images taken with an anamorphic lens (which changes the ratio of image height to width).

### structure

Define horizontal and vertical lines in the image using a manual or automatic method (see workflow section for details).

### fit

Set the distortion correction sliders automatically based on the identified structure (see workflow section for details).

### show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See [guides & overlays](#) for details.

## examples

Here is an image with a skewed horizon and converging lines caused by directing the camera upwards:



Here is the image after having corrected for vertical and horizontal perspective distortions using automatic structure detection. Note the framing adjustment made by the automatic cropping feature and the still-visible overlay of structural lines:



## 8.2.59. rotate pixels

The sensors of some cameras (such as the Fujifilm FinePix S2Pro, F700, and E550) have a diagonally oriented Bayer pattern instead of the usual orthogonal layout.

Without correction this would lead to a tilted image with black corners. This module applies the required rotation.

darktable detects images that need correction using their Exif data and automatically activates this module where required. For other images the module always remains disabled.

The module has no controls.

## 8.2.60. scale pixels

Some cameras (such as the Nikon D1X) have rectangular instead of the usual square sensor cells. Without correction this would lead to distorted images. This module applies the required scaling.

darktable detects images that need correction using their Exif data and automatically activates this module where required. For other images the module always remains disabled.

The module has no controls.

## 8.2.61. shadows and highlights

Modify the tonal range of the shadows and highlights of an image by enhancing local contrast.

---

**Note:** *This module performs blurs in Lab color space, which can result in a number of issues, including halos, high local contrast in highlights, and hue shifts towards blue in the shadows. Instead, use the [tone equalizer](#) module instead.*

---

## module controls

### shadows

Control the effect on shadows. Positive values will lighten shadows while negative values will darken them.

### highlights

Control the effect on highlights. Positive values will lighten highlights while negative values will darken them.

### white point adjustment

By default the algorithm of this module leaves the black and white points untouched. In some cases an image might contain tonal variations beyond the white point, i.e. above a luminance value of 100. A negative shift in the white point adjustment can bring these values back into the proper range so that further details in the highlights become visible.

### soften with

The underlying blurring filter: gaussian or bilateral. Try the bilateral filter if the gaussian blur generates halos.

### radius

The radius of the blurring filter used by the algorithm. Higher values give softer transitions between shadows and highlights but might introduce halos. Lower values will reduce the size of halos but may lead to an artificial look. The bilateral filter is much less prone to halo artifacts.

### compress

Control how strongly the effect extends to the mid-tones. High values limit the effect to only the extreme shadows and highlights. Lower values also cause adjustments to the mid-tones. At 100% this module has no visible effect as only absolute black and absolute white are affected.

### shadows color adjustment

Control the color saturation adjustment made to shadows. High values cause saturation enhancements on lightened shadows. Low values cause desaturation on lightened shadows. It is normally safe to leave this at its default of 100%. This gives a natural saturation boost on shadows – similar to what you would expect in nature if shadows were to receive more light.



### highlights color adjustment

Control the color saturation adjustment made to highlights. High values cause saturation enhancements on darkened highlights. Low values cause desaturation on darkened highlights. Often highlights do not contain enough color information to give convincing colors when darkened. You might need to adjust this parameter in order to find the best fitting value depending on your specific image, but be aware that sometimes the results might not be fully satisfying.

## 8.2.62. sharpen

Sharpen the details in the image using a standard UnSharp Mask (USM).

This module works by increasing the contrast around edges and thereby enhancing the *impression* of sharpness of an image. This module is applied to the L channel in Lab color space.

---

**Note:** *The USM algorithm used in this module performs blurs in Lab color space, which can produce undesirable effects, and is no longer recommended. Instead use the presets offered by the [contrast equalizer](#) module for deblurring or the [local contrast](#) module for general sharpness.*

---

## module controls

### radius

The unsharp mask applies a gaussian blur to the image as part of its algorithm. This parameter controls the radius of that blur which, in turn, defines the spatial extent of the edge enhancement. Very high values will lead to ugly over-sharpening.

### amount

The strength of the sharpening.

### threshold

Contrast differences below this threshold are excluded from sharpening. Use this to avoid amplification of noise.

## 8.2.63. soften

Create a softened image using the [Orton effect](#).

Michael Orton achieved his results on slide film by using two exposures of the same scene, one well exposed and one overexposed. He then used a darkroom technique to blend these two images into a final image where the overexposed image was blurred.

This module is a near-copy of Orton's analog process into the digital domain.

## module controls

### size

The size of blur of the overexposed image. The bigger the size, the softer the result.

### saturation

The saturation of the overexposed image.

### brightness

The brightness (EV) of the overexposed image.

### mix

How the two images are mixed. 50% represents an equal mix of the well exposed and overexposed images.

## 8.2.64. split-toning

Create a two color linear toning effect where the shadows and highlights are represented by two different colors.

The split-toning module does not convert images to black and white and has limited benefits on color images. In order to perform traditional split-toning, the input to this module should therefore be monochrome.

## module controls

### shadows and highlights color

Set the desired hue and saturation for both shadows and highlights. Clicking on the colored squares will open a color selector dialog which offers you a choice of commonly used colors, or allows you to define a color in RGB color space.

### balance

The ratio of toning between shadows and highlights. When set to 50%, half of the lightness range in image is used for shadows toning and the other half for highlights toning.

### compress

The percentage of total (mid-tone) lightness range that is not affected by color toning. This compresses the effect on the shadows and highlights while preserving the mid-tones.

## 8.2.65. spot removal (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.6 and should no longer be used for new edits. Please use the “cloning” tool in the [retouch](#) module instead.**

---

Correct areas of an image (the target) using details from another area of the image (the source).

This module uses some of the shapes that are available for [drawn masks](#) (circles, ellipses and paths), and the user interface is similar.

To use this module, first select the desired shape by clicking the corresponding shape icon (Ctrl+click to enable continuous shape creation mode). Next, you may optionally choose the source for the correction by using Shift+Ctrl+click or Shift+click on the image canvas to choose absolute or relative mode, respectively (see below). Finally click on the canvas to choose the area to be healed (the target area). A cross is displayed where the source area will be positioned.

The source positioning has two modes

### absolute mode

Before placing a shape on the image canvas, Shift+Ctrl+click on the desired position for the source. From this point on, all new shapes will have their source created at the same absolute position (in image co-ordinates).

### relative mode

Before placing a shape on the image canvas, Shift+click on the desired position for the source. The current shape will have the source created at that position and subsequent shapes will have their source created at the same position, *relative to the target shape*.

After creation the source and target areas of each shape can be shifted independently until the result matches your expectations. An arrow points from the source area of each shape to its target.

Use the shape-specific controls to adjust each shape’s size, border width, and other attributes.

Right-click on a shape to delete it.

This module is equivalent to the “cloning” tool in the [retouch](#) module; see the description of the cloning tool for additional details and examples. If *spot removal* produces unsatisfactory results, you may want to try the “heal” tool in *retouch* instead.



## 8.2.66. surface blur

Smooth image surfaces while preserving sharp edges using a bilateral filter.

This module *can* be used to denoise images, however you should be aware that bilateral filters are susceptible to overshoots and darktable offers much better alternatives. For example, the [astrophoto denoise](#) module uses a non-local means denoising algorithm, and the [denoise \(profiled\)](#) module provides a choice between non-local means and wavelet denoising algorithms.

The *surface blur* module blurs noise within the surfaces of an image by averaging pixels with their neighbors, taking into account not only their geometric distance but also their distance on the range scale (i.e. differences in their RGB values). It can be particularly useful if one RGB channel is more noisy/needs more smoothing than the others. In such a case, use the [color calibration](#) module to examine the channels one by one, in order to set the blur intensities accordingly.

The module can also be used as a creative filter to provide interesting effects, for example to lend an image a cartoon-like appearance. When used with chrominance blending, it can also be used to even out the color of a surface (for example, to remove skin redness).

This module is resource-intensive and slows down pixelpipe processing significantly. You should consider activating it late in your workflow.

### module controls

#### radius

The spacial extent of the gaussian blur.

#### red, green, blue

The blur intensity for each of the RGB channels.

## 8.2.67. tone curve

A classic digital photography tool to alter the tones of an image's using curves.

This module is very similar to the [rgb curve](#) module but works in Lab color space.

Activate the color picker to show the picked values on the displayed histogram (Ctrl+click or right-click to use the picker in area mode). Numerical (Lab) values of the input and output (see below) at the selected spot or area are shown at the top left of the graph.

### module controls

Please refer to the [curves](#) section for more details about how to modify curves including the **interpolation method**, **scale for graph** and **preserve colors** controls.

## color spaces

Depending on the desired intent, the tone curve can be applied in one of three color spaces:

- *Lab (linked or separated channels)*: Lab is a perceptual color space that is designed to approximate the way humans perceive colors and lightness. It represents color information independently of lightness. In “Lab, separated channels” mode, you are given fully independent control over the luminance (L-channel) and the chrominance (a/b-channels). In “Lab, linked channels” mode, only the luminance (L-channel) control is available. The color saturation correction is automatically computed, for each pixel, from the contrast correction applied to the luminance channel. This works better in cases where a subtle contrast correction is applied, but gives increasingly inaccurate saturation correction as the contrast is more dramatically altered.
- *XYZ (linked channels)*: XYZ is a linear technical color space designed to link the physiological light response of the human eye to RGB color spaces. As with Lab, it separates lightness from color information, but does so in a way that does not account for the role of the brain’s correction in human perception. The “XYZ, linked channels” mode offers an alternative to “Lab, linked channels”, and works by applying the L-channel curve to all three channels in the XYZ color space. Look at [blend mode “coloradjustment”](#) if you want to tune the strength of automatic chroma scaling. This mode is known to produce a slight hue shift towards yellow.
- *RGB (linked channels)*: RGB is a linear color space designed to capture and display images in additive synthesis. It is related to capture and display media and does not isolate color and lightness information. The “RGB, linked channels” mode works in ProPhoto RGB and applies the L-channel curve to all three channels in the RGB color space. Adding contrast in RGB space is known to desaturate highlights and boost saturation in the shadows, but this has proven to be the most reliable way to edit contrast, and is the standard method used in most software. This mode makes the tone curve module behave in much the same way as the [base-curve](#) module, except that the latter works in camera RGB space.

Note that the module works in Lab color space in all cases. This means that the middle-gray coordinate is always 50% in the graph, no matter what color space is used. The same applies to the inset histogram displayed in the background of the curve. The controls are converted to the relevant color space before the corrections are applied – in RGB and XYZ, the middle-gray is therefore remapped from 50% to 18%.

## L-channel curve

The tone curve in L-channel works on Lightness. To provide a better overview, a lightness histogram is displayed in the module. When working in one of the “linked channels” modes, the L-channel curve is the only one available.

The horizontal axis represents the lightness of the input image pixels. The vertical axis represents the lightness of the output image pixels. The default straight diagonal line does not change anything. A curve above the default diagonal increases the lightness, whereas a curve below decreases it. An S-like curve will enhance the contrast of the image. You can move the end-points of the default diagonal to change the black and white points and hence alter the overall contrast – this has the same effect as changing the black and white points in the [levels](#) module.

## a/b-channel curves

The curves in the a and b channels work on color values and are available only in the “Lab, separated channels” mode. The horizontal axis represents the color channel value of the input image pixels. The vertical axis represents the color channel value of the output image pixels. Positive a-values correspond to more magenta colors; negative a-values correspond to more greenish colors. Positive b-values correspond to more yellowish colors; negative b-values correspond to more blueish colors.

The default diagonal straight line does not change anything. Shifting the center of the curve will give the image a color tint: shifting the a-channel upwards gives a magenta tint; shifting the b-channel upwards gives a yellow tint; shifting the a-channel downwards gives a green tint; shifting the b-channel downwards gives a blue tint.

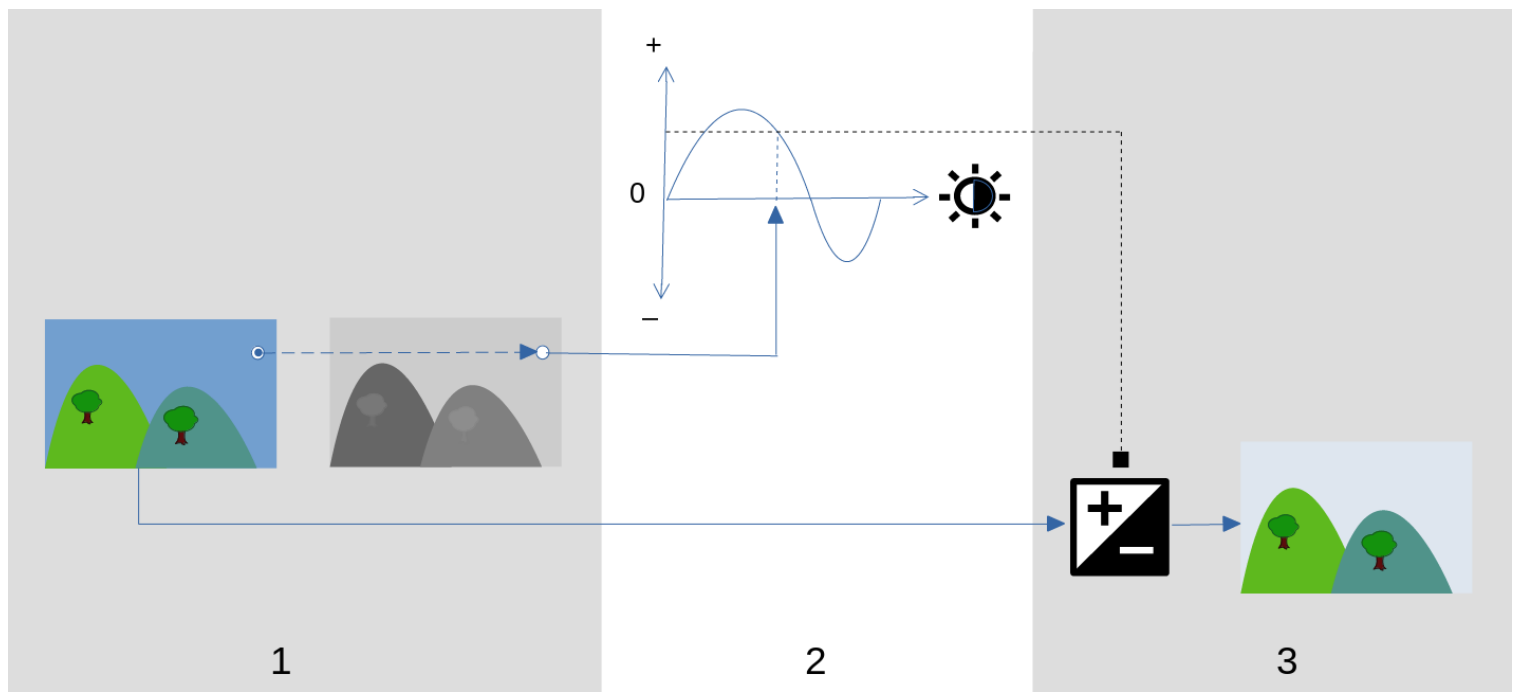
Increasing/decreasing the steepness of a curve, without shifting its center, will increase/decrease the color saturation of the respective channel. With properly defined curves you can exert fine control on color saturation, depending on the input pixel’s colors.

## 8.2.68. tone equalizer

Dodge and burn while preserving local contrast.

When used together with [filmic rgb](#), this module replaces the need for other tone-mapping modules such as the [base curve](#), [shadows and highlights](#), [tone curve](#) and [zone system \(deprecated\)](#) modules. It works in linear RGB space and utilizes a user-defined mask to guide the dodging and burning adjustments, helping to preserve local contrast within the image.

The following diagram describes how the tone equalizer works:



1. Create a monochrome [guided mask](#) that divides the input image into regions of similar luminosity. The resulting mask should blur the fine details within the image so that pixels within each region are all treated similarly, preserving local contrast.
2. Adjust the sliders in the [simple](#) tab or the equalizer graph in the [advanced](#) tab to alter the brightness of the underlying image, based on the brightness of the mask. Exposure can also be adjusted by scrolling with the mouse while hovering the cursor over the preview image (see [cursor indicator/control](#) for details).

In the simple tab, each slider corresponds to a single brightness zone (EV) in the mask, which can be raised or lowered to adjust the exposure of the image where the mask's brightness lies in that zone. Similarly, in the equalizer tab, the horizontal axis of the graph corresponds to the brightness level of the mask, and the vertical axis specifies the required exposure adjustment to pixels where the mask matches that brightness level.

3. The exposure of each pixel of the input image is adjusted using the mask and the equalizer graph. For each pixel, the module looks up the brightness of the mask at that point, finds the matching brightness on the horizontal axis of the equalizer graph, and increases or decreases the exposure accordingly (using the vertical axis of the graph).

It is important that the mask separates the image into regions of similar brightness, and that a suitable amount of blur is applied within those regions. This means that all of the pixels in each region will have their exposure adjusted similarly, without adversely affecting local contrast. Examine your image beforehand to identify which regions you wish to dodge and burn, and use the controls on the [masking](#) tab to ensure that those areas are reasonably separated in tone within the final mask. This will allow those regions to be adjusted independently.

## module controls

The controls of the *tone equalizer* module are divided between three tabs.

### display exposure mask

Click on the icon to the right of this label to show/hide the module's guided mask over the top of the image. This control is available in all three tabs.

### simple tab

This tab splits the brightness of the guided mask into nine zones (from -8 to 0 EV) and allows you to alter each zone independently. This is a simplified interface and is used to generate the same tone adjustment curve as shown in the [advanced](#) tab.

## -8 EV ... 0 EV

Each slider adjusts the exposure of all pixels where the *guided mask* has the given brightness. If the mask's histogram is evenly spread over the entire tonal range, sliders towards the top generally affect the shadows, whereas sliders towards the bottom generally affect the highlights. You can check the spread of the histogram within the [advanced](#) tab.

## advanced tab

This tab allows you to control the same intensity levels as in the simple tab, though here they are represented as control points on a curve. Behind the curve is a histogram representing the intensity levels of the *guided mask* (not the underlying image). If the histogram is too bunched up, this means your mask doesn't have a good spread of intensity levels, which makes it harder to independently control the brightness of the different parts of your image. It is therefore recommended that the histogram be adjusted so that it extends the entire range, covering as many control points as possible for maximum flexibility. You can adjust the mask using the controls in the [masking](#) tab.

Click+drag the control points on the curve to adjust the brightness of all pixels where the mask has the given intensity. If the mask's histogram is evenly spread over the entire tonal range, control points to the left will generally affect the shadows, and control points to the right will generally affect the highlights. Moving a single control point will also affect control points on either side to ensure the curve remains smooth. This behaviour can be adjusted using the *curve smoothing* control.

### curve smoothing

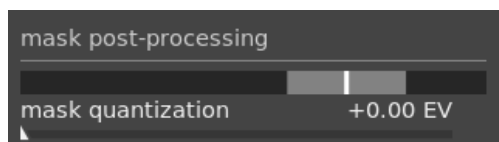
Control how the curve is interpolated between control points. Move the slider to the right to make the transitions between the control points more gradual, but beware that going past about 0.6 can introduce some instability (oscillations) in the curve due to mathematical constraints. Move the slider to the left for a more well-behaved curve, but beware that this can result in harsher tonal transitions that may damage local contrast.

## masking tab

This tab contains controls for adjusting the guided mask.

The purpose of the guided mask is to separate out areas with different tonal ranges so that they can be independently brightened or darkened by the tone equalizer. The masking filters are designed to allow sharp edges between these areas to be preserved, while blurring details within a given tonal range, so that the brightness can be adjusted without adversely impacting local contrast. Ideally the mask histogram shown in the *advanced* tab should be spread out across all of the control points.

To avoid having to switch back and forth between the *advanced* and *masking* tabs, a gray bar under the "mask post processing" label displays a representation of the middle 80% of the histogram. By using the controls in this tab to center and spread out this gray bar, you can expect to have a nicely shaped histogram when you return to the "advanced" tab. If you see orange at either end of the gray bar, this means that part of the histogram is outside of the 9 EV range of the mask, and needs to be further adjusted.



When setting up the guided mask you will need to strike a balance between obtaining a smooth blur within tonal regions (to preserve local contrast) and preservation of the boundaries between those regions. Some experimentation will be required to find the best settings. Often the key controls to adjust are the *exposure/contrast compensation* sliders at the bottom of the module.

Displaying the guided mask while making these adjustments will help you to understand these controls and better judge the quality of the mask.

### luminance estimator

Choose the method by which the luminance of a pixel will be estimated when mapping it to a mask intensity value (default *RGB euclidean norm*).

## preserve details

Choose the smoothing algorithm used to blur the mask:

- *no*: Do not smooth the mask (the effect is the same as using normal tone curves). When the module is used to compress dynamic range, this option can cause compression of local contrast. This can be useful when increasing (local and global) contrast.
- *guided filter*: Use the original guided filter algorithm to blur the mask while attempting to preserve edges. One of the limitations of this algorithm is that the guided filter is exposure-sensitive, meaning that shadows tend to be blurred more than highlights. Note that this limitation can sometimes be an asset: if one wants to lighten shadows a lot, the guided filter can provide very good local contrast preservation.
- *average guided filter*: Use this option in cases where the effect of the guided filter is too strong. In this mode, a geometric mean is taken between the output of the original *guided filter* algorithm and the output given by the *no* option.
- *eigf (default)*: The *exposure-independent guided filter* solves the problem of the original *guided filter*, in that it makes the degree of blurring independent of the exposure. This means the degree of blurring applied to the highlights and the shadows regions should be about the same. This improved algorithm is now the default option.
- *averaged eigf*: This option takes the geometric mean between the *eigf* mask and the mask generated by the *no* option, and is useful in cases where the degree of blurring in the mask needs to be mitigated.

## filter diffusion

By default this is set to a value of 1, meaning that the filtering algorithm is run once on the input image to produce the blurred monochrome mask.

If you increase this to 2, the filtering algorithm will be run once on the input image to produce an intermediate mask, and then a second time on the intermediate mask. As a result, the final mask will be more blurred than when using a single iteration. Progressively higher values will diffuse the mask even further. However, because the mask filtering algorithm is run multiple times, each iteration will increase the processing time.

## smoothing diameter

This controls how much of the surrounding image to take into account when calculating the mask's blur at a particular point, defined as a percentage of the length of the image's longer side (default 5%). With lower values, the transition between darker and lighter areas of the mask will be more pronounced. As the value is increased, transitions become smoother/softer. For the default *exposure-independent guided filter (eigf)*, you should typically use blurring radii around 1-10%. With the original *guided filter*, blurring radii around 1-25% will typically yield better results.

## edges refinement/feathering

Higher values force the mask to follow high contrast edges more closely. Lower values give smoother gradients, but may introduce halos. If required, feathering can be set to values as high as 10,000.

## mask post-processing

This bar provides a representation of the current span of the mask's histogram. It covers the middle 80% of the histogram, dropping the first and last decile to prevent outliers from skewing the indicator too much. Orange indicators at either end mean that the histogram exceeds the upper or lower bounds of its 9 EV range.

## mask quantisation

Apply a degree of posterisation to the mask, so that it tends to centre round a few discrete levels. In some cases, this may be useful to help separate out areas of your image into distinct masking levels.

## mask exposure compensation

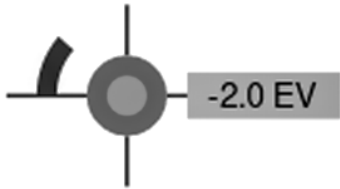
Adjust the mask's histogram to the left or right. If you have used the *exposure* module to adjust the image brightness, you may need to offset that adjustment by using this slider to re-centre the mask's histogram. Click on the wand icon to the right of the slider to set the exposure compensation such that the average of the mask's histogram will coincide with the central -4EV control point. The slider can then be fine-tuned as required.

## mask contrast compensation

Dilate (spread out) or compress the mask's histogram. The wand icon to the right of the slider will propose a reasonable starting point, which can then be fine-tuned to optimize the spread of the histogram under the tone equalizer control points.

## cursor indicator/control

When the *tone equalizer* module is enabled and expanded, you can move the mouse pointer over the preview image to show a cursor that displays information about the pixel under the pointer. When this cursor is shown, the mouse wheel can be used to brighten or darken the areas of your image that match the mask intensity level at that point. This provides a convenient way to quickly brighten or darken specific parts of the image.



- the cross-hairs indicate the position of the pixel under the cursor
- the text label shows the intensity of the guided mask at that point, in EV
- the shade of the outer circle indicates the intensity of the mask at that point
- if the tone equalizer has brightened or darkened pixels matching this mask intensity, the magnitude of the adjustment is indicated by an arc on the left-hand-side. The longer the arc, the greater the brightness adjustment,
- if there has been an exposure adjustment, the shade of the inner circle indicates the amount of brightening or darkening, relative to the mask's intensity at that point (as indicated by the outer gray circle). That is to say, if the pixel under the crosshairs has been brightened, the inner circle will be a lighter shade of gray than the outer circle; if the pixel has been darkened, the inner circle will be a darker shade of gray than the outer circle.

## presets

The *tone equalizer* comes with several presets that can be used to compress shadows and highlights. Each comes in two variants, using either the guided filter (gf) or the exposure-independent guided filter (eigf). The variants using the guided filter tend to preserve local contrast in the shadows better than those that use the exposure-independent guided filter, but at the price of reducing the local contrast in the highlights. Either of these variants may lead to better results, depending on the image. In both cases, the presets preserve middle-gray, so you shouldn't need to adjust the global exposure after activating the tone equalizer.

## 8.2.69. tone mapping (deprecated)

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [tone equalizer](#) module instead.**

Compress the tonal range of HDR images so that they fit into the limits of an LDR image, using Durand's 2002 algorithm.

The underlying algorithm uses a bilateral filter to decompose an image into a coarse base layer and a detail layer. The contrast of the base layer is compressed, while the detail layer is preserved, then both layers are re-combined.

## module controls

### contrast compression

The contrast compression level of the base layer. A higher compression will make the image fit a lower dynamic range.

### spatial extent

The spatial extent of the bilateral filter. Lower values cause the contrast compression to have stronger effects on image details.

## 8.2.70. unbreak input profile

Add a correction curve to image data. This is required if you have selected certain input profiles in the [input color profile](#) module.

If you decide to use an ICC profile from the camera manufacturer in the [input color profile](#) module, a correction curve frequently needs to be pre-applied to image data to prevent the final output from looking too dark. This extra processing is not required if you use darktable's standard or enhanced color matrices.

The correction curve is defined with a linear part extending from the shadows to some upper limit and a gamma curve covering mid-tones and highlights. For further information please see darktable's neighboring project [UFRaw](#).

### linear

The upper limit for the region counted as shadows and where no gamma correction is performed. Typically values between 0.0 and 0.1 are required by the profile.

---

**gamma**

The gamma value required to compensate the input profile. Often the required value is 0.45 (the reciprocal of the 2.2 gamma used by some manufacturer's profiles).

## 8.2.71. velvia

Resaturate pixels in a weighted manner that gives more weight to blacks, whites and less saturated pixels.

---

**Note:** *This module causes hue and brightness shifts that can be difficult to manage. Instead, use the [color balance rgb](#) module for color modifications.*

---

## module controls

**strength**

The strength of the effect

**mid-tones bias**

Reduce the effect on mid-tones in order to avoid unnatural skin tones. Reducing the mid-tone bias decreases mid-tone protection and makes the overall velvia effect stronger.

## 8.2.72. vibrance (deprecated)

---

**Please note that this module is [deprecated](#) from darktable 3.6 and should no longer be used for new edits. Please use the vibrance control in the [color balance rgb](#) module instead.**

---

Saturate and reduce the lightness of the most saturated pixels to make the colors more vivid.

## module controls

**vibrance**

The amount of vibrance to apply to the image.

## 8.2.73. vignetting

Apply a vignetting effect to the image.

Vignetting is a modification of the brightness and saturation at the borders of the image in a specified shape. Many of the parameters listed below can also be modified with a graphical control that overlays the image when the module has focus, showing the shape and extent of the effect.

---

**Note:** This module is known to provoke banding artifacts under certain conditions. You should consider activating the [dithering](#) module to alleviate this.

---

**Also note:** *This module can produce unnatural-looking results and should be used with care. Instead, use the [exposure](#) module with an [elliptical mask](#) with a large transition area and, if necessary, use the [color balance rgb](#) module with the same mask to reduce saturation at the edges.*

---

## module controls

**scale**

The radius of the vignetting area.

**fall-off strength**

The progressiveness of the fall-off. Higher values cause a steeper transition.

**brightness**

The intensity of brightening (positive values) or darkening (negative values).

---



**saturation**

Control how strong colors become when desaturated/saturated in the darkened/brightened vignetting area.

**horizontal/vertical center**

Shift the center of the vignetting area horizontally/vertically.

**shape**

The shape of the vignetting effect. The default value of 1 creates a circular or elliptical area. Smaller values shift the shape to being more square; higher values turn it into a cross-like shape.

**automatic ratio**

Automatically adjust the width/height ratio of the vignetting area to match the aspect ratio of the underlying image. This typically causes the vignetting area to become elliptical.

**width/height ratio**

Manually adjust the width/height ratio of the vignetting area.

**dithering**

Activate random noise dithering to alleviate banding artifacts caused by vignette gradients. Select “8-bit output” to prevent banding on monitor display and for JPEGs. When set to “16-bit output”, only a small amount of dithering is applied, just strong enough to compensate for banding on the fine grained 16-bit level. It is now recommended that you instead use the [dithering](#) module to alleviate banding artifacts.

## 8.2.74. watermark

Render a vector-based overlay onto your image. Watermarks are standard SVG documents and can be designed using [Inkscape](#).

You can also use bitmap (PNG) images.

The SVG processor in darktable can also substitute strings within an SVG document, allowing image-dependent information to be included in the watermark.

User-designed watermarks should be placed into the directory `$HOME/.config/darktable/watermarks`. Once in place, use the reload button update the list of available watermarks.

The following is a list of variable strings that are supported for substitution within an SVG document.

In addition to this list you can also use the variable strings defined in the [variables](#) section.

<code>\$(DARKTABLE.NAME)</code>	The application name
<code>\$(DARKTABLE.VERSION)</code>	The version of darktable
<code>\$(WATERMARK_TEXT)</code>	A short free text (max. 63 characters)
<code>\$(WATERMARK_COLOR)</code>	The color to use for <code>\$WATERMARK_TEXT</code>
<code>\$(WATERMARK_FONT_FAMILY)</code>	The font family to use for <code>\$WATERMARK_TEXT</code>
<code>\$(WATERMARK_FONT_STYLE)</code>	The font style (normal, oblique, italic)
<code>\$(WATERMARK_FONT_WEIGHT)</code>	The font weight (boldness)
<code>\$(IMAGE.ID)</code>	The unique image id within current library
<code>\$(IMAGE.FILENAME)</code>	The image filename
<code>\$(IMAGE.BASENAME)</code>	The image file basename
<code>\$(IMAGE.EXIF)</code>	The image Exif string
<code>\$(EXIF.DATE)</code>	The image date
<code>\$(EXIF.DATE.SECOND)</code>	Seconds from the image Exif data
<code>\$(EXIF.DATE.MINUTE)</code>	Minutes from the image Exif data
<code>\$(EXIF.DATE.HOUR)</code>	Hours from the image Exif data (24h)
<code>\$(EXIF.DATE.HOUR_AMPM)</code>	Hours from the image Exif data (12h, AM/PM)
<code>\$(EXIF.DATE.DAY)</code>	Day of month from the image Exif data (01 .. 31)
<code>\$(EXIF.DATE.MONTH)</code>	Month from the image Exif data (01 .. 12)
<code>\$(EXIF.DATE.SHORT_MONTH)</code>	Month from the image Exif data localized (Jan, Feb, .. Dec)
<code>\$(EXIF.DATE.LONG_MONTH)</code>	Month from the image Exif data localized (January, February, .. December)
<code>\$(EXIF.DATE.SHORT_YEAR)</code>	Abbreviated year from the image Exif data (2013 is "13")
<code>\$(EXIF.DATE.LONG_YEAR)</code>	Full year from the image Exif data
<code>\$(DATE)</code>	Current system date
<code>\$(DATE.SECOND)</code>	Current system time seconds
<code>\$(DATE.MINUTE)</code>	Current system time minutes
<code>\$(DATE.HOUR)</code>	Current system time hours (24h)
<code>\$(DATE.HOUR_AMPM)</code>	Current system time hours (12, AP/PM)
<code>\$(DATE.DAY)</code>	Current system time day of month (01 .. 31)

---

<code>\$(DATE.MONTH)</code>	Current system time month (01 .. 12)
<code>\$(DATE.SHORT_MONTH)</code>	Current system time month localized (Jan, Feb, .. Dec)
<code>\$(DATE.LONG_MONTH)</code>	Current system time month localized (January, February, .. December)
<code>\$(DATE.SHORT_YEAR)</code>	Current system time year (abbreviated)
<code>\$(DATE.LONG_YEAR)</code>	Current system time year
<code>\$(EXIF.MAKER)</code>	The maker of the camera
<code>\$(EXIF.MODEL)</code>	The camera model
<code>\$(EXIF.LENS)</code>	The specific lens used
<code>\$(Xmp.dc.creator)</code>	The creator string
<code>\$(Xmp.dc.publisher)</code>	The publisher string
<code>\$(Xmp.dc.title)</code>	The title of the image
<code>\$(Xmp.dc.description)</code>	The description of the image. Some cameras use this to store shoot-time user-specified text (also called “Exif user comment”)
<code>\$(Xmp.dc.rights)</code>	The rights assigned to the image
<code>\$(GPS.LATITUDE)</code>	The image latitude coordinate (N/S 0 .. 90)
<code>\$(GPS.LONGITUDE)</code>	The image longitude coordinate (E/W 0 .. 180)
<code>\$(GPS.ELEVATION)</code>	The image elevation level (meters)
<code>\$(GPS.LOCATION)</code>	All three coordinates (latitude, longitude, elevation)

## module controls

### marker

Choose the watermark to apply. Use the reload button to update the list to include any newly-added watermarks. The extension of the file (png/svg) is shown in brackets.

### text

A free text field in which you can enter up to 63 characters to be printed where referenced by the corresponding watermark. An example is supplied as `simple-text.svg`.

### text color

The color of the text. Click on the colored field to open a color selector dialog which offers you a choice of commonly used colors, or allows you to define a color in RGB color space.

### text font

The font to use (default “DejaVu Sans Book”). Click on the field to open a dialog box showing the fonts available on your system. Fonts can be searched by name and a preview is shown next to the font name. You may specify your own sample text.

### opacity

The opacity of the watermark’s rendering.

### scale

The scale of the watermark, with respect to the option selected in the “scale on” parameter.

### rotate

The rotation angle of the watermark.

### scale on

The reference for the scale parameter. The default setting “image” scales the watermark relative to the horizontal image size. Alternatively you can scale the watermark relative to the “larger border” or “smaller border”.

### alignment

Use these controls to align the watermark to any edge or the center of the image.

### x offset

Pixel-independent offset relative to the choice of alignment on the x-axis.

### y offset

Pixel-independent offset relative to the choice of alignment on the y-axis.

## 8.2.75. white balance

Adjust the white balance of the image by altering the temperature and tint, defining a coefficient for each RGB channel, or choosing from list of predefined white balance settings.

The default settings for this module are derived from the camera white balance stored in the image’s Exif data.

White balance is not intended as a “creative” module – its primary goal is to technically correct the white balance of the image ensuring that neutral colored objects in the scene are rendered with neutral colors in the image. For creative color operations, it is usually better to use other modules such as [color calibration](#) or [color balance rgb](#).

---

**Note:** The [color calibration](#) module now provides a more modern and flexible method of controlling white balance. The *color calibration* module can be enabled by default for new images by selecting “modern” in [preferences > processing > auto-apply chromatic adaptation defaults](#). Some basic settings are still required (and applied automatically) in the *white balance* module, so that [demosaic](#) works correctly.

---

## module controls

### scene illuminant temp

This section provides scene-illuminant *temperature* and *tint* controls to adjust the white balance of the image. Click on the ‘scene illuminant temp’ section label to cycle between 3 different [color modes](#) for the temperature/tint sliders.

#### temperature

Set the color temperature in kelvin.

#### tint

Alter the color tint of the image, from magenta (tint < 1) to green (tint > 1).

### white balance presets

#### setting

Choose from a predetermined list of white balances. The available settings are derived from the presets available in the camera used to take the photograph. The following options are provided in addition to any camera-defined white balance presets.

- *as shot* (default): The white balance as reported by the camera
- *from image area*: Draw a rectangle over a neutral color in the image to calculate white balance from that area.
- *user modified*: The most recently modified setting. Manual adjustment of temperature, tint or r/g/b channel coefficients will automatically select this option. Choose this setting after selecting any other preset to return parameters to the most recent user-modified state
- *camera reference*: Set the temperature to the camera reference white point, which is assumed to be D65 (or ~6502K). The white balance channel multipliers are calculated such that pure white in the camera colorspace is converted into pure white in sRGB D65 (where pure white means that each color channel has an equal value).

For convenience the final four modes can also be set by clicking on one of the buttons in the [button bar](#) above the *setting* drop-down.

#### finetune

Finetune a camera-specific white balance preset. This is only shown if it is available for the camera in question. The direction of adjustment is dependent on the provided presets. If your camera doesn’t have white balance presets available, check [this guide](#) to see how you can submit your own.

### channel coefficients

The RGB channel coefficients are automatically calculated from the above parameters and, as such, are hidden by default. You can expand/collapse the channel coefficients section by clicking on either the ‘channel coefficients’ label or the adjacent triangular button.

#### red/green/blue

Set the value of each RGB channel coefficient from 0 to 8

### additional functionality

#### colored sliders

By default the module’s sliders are monochrome. However, two flavors of colored sliders can be enabled in [preferences > darkroom > white balance slider colors](#) or by clicking on the ‘scene illuminant temp’ section label in the module.

#### no color (default)

The background of the sliders is not colored.

**illuminant color**

The slider colors represent the color of the light source (the color you are adjusting to in order to achieve neutral white).

**effect emulation**

The slider colors represent the effect the adjustment would have had on the scene. This is how most other raw processors show temperature/tint sliders colors.

**button bar**

The button bar is simple addition that allows one-click access to the internal white balance settings. You can disable this by editing your `darktable.rc` file. Find the line that says

```
plugins/darkroom/temperature/button_bar=TRUE
```

and change TRUE to FALSE.

**usage warning**

The only parameters that are used internally by this module's operation are the *rgb channel coefficients*. The *temperature* and *tint* sliders are provided as a more user-friendly way to adjust those parameters. The relationship between the channel coefficients and temperature/tint sliders depends on characteristics specific to the camera used to take the photograph. This means that applying white balance settings from an image made with one camera to an image made with another will, in general, not give consistent results.

The mathematical relationship between the two sets of values is not straightforward. It is possible to set the channel coefficients such that there is no valid equivalent temperature and tint setting (mainly where very high temperature values are calculated from the sliders). Editing white balance using temperature and tint on an image previously edited using channel coefficients may therefore give odd results, at least where high temperature values are involved.

**8.2.76. zone system (deprecated)**


---

**Please note that this module is [deprecated](#) from darktable 3.4 and should no longer be used for new edits. Please use the [tone equalizer](#) or multiple instances of the [exposure](#) module with [parametric masks](#) to narrow down on a zone.**

---

Adjust the lightness of an image using Ansel Adams' [zone system](#) .

The lightness of the image (the L channel in Lab color space) is divided into a number of zones ranging from pure black to pure white. These zones are displayed in a zonebar beneath a preview image. The number of zones can be changed by mouse-scrolling on that bar (default 10 zones).



The zonebar is split horizontally with the lower part showing the zones of the module's input and the upper part the zones of the module's output. In its default state both parts are fully aligned. While the output zones are static you can left click and drag a control point in the input zones to modify the mapping between input and output.

Shifting a control point proportionally expands the zones on one side and compresses the zones on the other side of that point. All pre-existing control points stay in place, effectively preventing changes to the zones beyond the next control point. Use right click to remove a control point.

The preview image above the zonebar shows how the image's zones are broken down. When hovering above a zone, that zone – either from input or output – is highlighted in yellow on the preview image.


## 8.3. utility modules

### 8.3.1. darkroom

#### 8.3.1.1. clipping warning

Highlight areas of the image that may exhibit luminance or gamut clipping.

When an image is sent to a display device, each pixel is normally represented as a set of 3 numbers, representing the intensity of the red, green and blue primary colors in the output color space. Because the output color space is usually closely related to hardware with physical limitations, there is a maximum permitted value for the [R,G,B] channels, representing the maximum available intensity for that color space. Similarly, there is also a minimum value below which pixel values will be mapped to zero. When we try to convert from a larger color space to the final output color space, any values exceeding this maximum will be clamped to the maximum value, and any values below the minimum will be clamped to zero. This process is called “clipping” and it will lead to lost detail, or “incorrect” colors for any pixels with clipped channels.

Click the  icon to enable the clipping warning.

There are two ways in which a pixel might become clipped when represented in the output color space.

- *luminance clipping*: This can occur when a pixel is too bright to be represented in the output color space. The pixel luminance is calculated as a weighted average of the [R,G,B] channels. If this average exceeds the maximum allowed value, it is an indication of over-exposure. The overall luminance of a pixel can also be too dark to be represented by an [R,G,B] value in the output color space, in which case it will simply be shown as black. We normally deal with luminance clipping by carefully adjusting tone mappings and exposure levels.
- *gamut clipping*: The output color space defines a set of primary colors that, mixed together in certain ratios, produce the final output color. However, there are only so many colors that can be produced by mixing together a combination of those three primary colors. Highly saturated colors in particular can be difficult to represent, especially for pixels that are very bright or very dark. If there is no set of positive [R,G,B] values that can represent a given color at a given level of brightness, we say that the color is “out of gamut”, and we need to settle for another color instead that *can* be represented by permitted [R,G,B] values within the color space. We can handle gamut clipping by being careful not to over-saturate colors in the highlights and shadows, and possibly by using some color grading/color mapping techniques.

The “clipping warning” module is used to highlight those pixels that cannot be accurately represented in the output color space, either due to luminance or gamut clipping. Prior to darktable 3.4, the clipping highlighted any pixels that exceeded the maximum allowed value on any of the [R,G,B] channels, or that had been completely crushed to black. From darktable 3.4 onwards, the clipping warning indicator has some additional modes to help you to differentiate between luminance and gamut clipping, so that you can make better decisions about how to address any issues.

As the clipping warning runs at the end of the preview pipeline, it receives data in display color space then converts it to histogram color space. If you are using a display color space that is not “well behaved” (this is common for a device profile), then colors that are outside of the gamut of the display profile will clip or distort.

The clipping warning module, described here, warns you about clipping caused by image processing and the limitations of the output color space. It should not be confused with the following similar tools:

- The [raw overexposed warning](#) indicates where pixels in the original raw file are clipped due to physical limitations in the dynamic range of the camera sensor. This module highlights information that was permanently lost at the point of image capture, and you need to deal with it as best you can using highlight recovery techniques.
- The [gamut check](#) module also provides information about clipping arising from image processing. It is based on the external littleCMS library, and is more or less equivalent to the *full gamut* mode in the clipping warning module. The downsides of the gamut check module are that it doesn’t allow you to distinguish between clipping caused by luminance and gamut mapping, and it is much slower than the clipping warning indicator.

#### module controls

Right-click on the clipping icon to show the following options:

### clipping preview mode

Choose the type of clipping that you want to highlight:

- *any RGB channel*: Provides an over-clipping indication if any one of the three [R,G,B] channels exceeds the maximum permitted value for the histogram color space, or an under-clipping indication if the three [R,G,B] channels are too dark and are all forced to black. This was the default mode prior to darktable version 3.4.
- *luminance only*: Indicates any pixels that are clipped because their luminance falls outside of the range set in the “upper threshold” and “lower threshold” sliders. If this happens, it generally means that tone mapping or exposure settings have been poorly set
- *saturation only*: Indicates where over-saturated colors have pushed one or more of the [R,G,B] channels towards a value outside the permitted range of the histogram color space, even though the overall luminance of the pixel may lie within acceptable limits. This means that the pixel’s color is impossible to represent in the histogram color space, and can arise from poorly set gamut mapping or saturation settings,
- *full gamut*: Shows a combination of the three previous options. This is the default mode from darktable 3.4 onwards, and it gives the most complete indication of potentially problematic pixels.

### color scheme

By default, the indicator marks pixels with *red* where the upper threshold is exceeded (over-clipping) and with *blue* where the lower threshold is breached (under-clipping). This color scheme can be changed to *black & white* or *purple & green* for “over & under” indicators, which may be useful to improve visibility for some images.

### lower threshold

Expressed in EV relative to the white point (which is nominally EV = 0). If the [R,G,B] channels all fall below this value, an under-clipping indicator is shown warning that the pixel may be end up being crushed to black. Use the following reference to set this threshold, depending on your intended output medium:

- *8-bit sRGB* clips blacks at -12.69 EV
- *8-bit Adobe RGB* clips blacks at -19.79 EV
- *16-bit sRGB* clips blacks at -20.69 EV
- *fine art matte prints* typically produce black at -5.30 EV
- *color glossy prints* typically produce black at -8.00 EV
- *black & white glossy prints* typically produce black at -9.00 EV

### upper threshold

How close a pixel should be to the upper limit before being flagged by the clipping warning, expressed as a percentage (default 98%). In the case of gamut checks, this controls how close the saturation of the pixel is allowed to get to the limits of the color space’s gamut before a clipping indication is flagged.

## 8.3.1.2. color assessment

Assess colors and brightness in your image using ISO 12646:2008 recommended viewing conditions.


When developing an image, the way we perceive brightness, contrast and saturation is influenced by the surrounding ambient conditions. If an image is displayed against a dark background, as with the default theme in darktable, this can have the a number of adverse effects on our perception of that image:

- Exaggeration of the perceived exposure makes the image seems brighter than it really is. This is nicely illustrated by the [Adelson checkerboard shadow effect](#) .
- A decrease in the perceived saturation in the image makes the colors seem less rich than they really are (the Hunt effect).
- A decrease in the perceived contrast in the image makes the tones seem flatter than they really are (Bartleson-Breneman effect 3)

The end result is that the final image can end up being too dark and overly-processed in terms of contrast and color saturation. To avoid this, the “ISO 12646:2008” standard makes some recommendations about the conditions under which the colors of an image should be assessed. The *color assessment* module in the darkroom places a frame around the image to help the user better assess the colors in the image, along the lines of those recommendations.





When the color assessment button  is selected in the bottom panel, the image is zoomed out so that a thick mid-gray border appears around the image to act as a reference against which to compare the image's tones. A thinner white border is placed immediately around the image to give the eyes a basis for comparison when looking at parts of the image that are meant to be a bright white.

Although the color assessment mode provides a mid-gray surrounding to the image, it is recommended that you also set your user interface (in [preferences > general](#)) to one of the “gray” themes. These themes are designed to provide a user interface that is close to middle gray (it is actually slightly darker to allow better contrast with the text in the user interface). When one of these themes is used together with the color assessment mode, this will help to avoid the above perception issues.

Color assessment mode can also be toggled by pressing Ctrl+B.

### 8.3.1.3. duplicate manager

View and create multiple versions of the current image. Each version can be edited independently without affecting other versions – all versions use the same underlying image file, but the editing history of each version is stored in its own independent XMP sidecar file.

The duplicate manager lists each version of the current darkroom image along with its preview thumbnail. Hold down the left mouse button on a thumbnail to temporarily show that version in the center view. Double-click to switch to that version and edit it.


The buttons at the bottom of the module allow you to create new duplicates of the current image. You can either create a ‘virgin’ version (with an empty history stack) using the “original” button or an exact duplicate of the current edit using the “duplicate” button.

A version number is displayed to the right of each thumbnail. Click in the area below the version number to enter a description against that version of the image. This description is stored in the *version name* metadata tag, which can also be edited within the [metadata editor](#) module in the lighttable view.



### 8.3.1.4. gamut check

Highlight areas of the image that may exhibit gamut clipping.

Click the  icon to activate the gamut check display mode for your image. This function highlights, in cyan, all pixels that are out of gamut with respect to the selected softproof profile. You can also activate gamut check with the keyboard shortcut Ctrl+G. A message “gamut check” on the bottom left of your image tells you that you are in gamut check display mode. Gamut check and [soft proof](#) are mutually exclusive modes.

Right-click on the icon to open a dialog with configuration parameters – these are the same as for the [soft proof](#) option.

You might also like to consider using the [clipping warning](#) , which also provides under- and over-exposure warnings as well as a gamut check similar to that offered by this module.

### 8.3.1.5. global color picker

Take color samples from the current darkroom image, display their values in multiple ways and compare colors from different locations.

The color picker is activated by pressing the color picker icon. The module’s parameters will remain in effect until you leave the darkroom mode.

Besides the global color picker described here, many darktable modules (e.g. [tone curve](#) ) also contain local color pickers which are used to set individual module parameters. You should be aware that these two forms of color picker do not always work in the same color space. The global color picker works in the histogram color space and takes its samples after the complete pixelpipe has been processed. Local color pickers run in the color space of the module in which they are activated and reflect the input or output data of that module within the pixelpipe.

You can right-click on the sampled color values to copy them to the clipboard.

As the global color picker runs at the end of the preview pixelpipe, it receives data in display color space then converts it to histogram color space. If you are using a display color space which is not “well behaved” (this is common for a device profile), then colors that are outside of the gamut of the display profile will clip or distort.

#### module controls

##### point/area mode

The global color picker can be activated in point or area mode by clicking or Ctrl+clicking on the color picker icon, respectively (you can also right-click to enable area mode). In point mode only a small spot under your cursor is taken as a sample. In area mode darktable samples the area within a drawn rectangle.

##### mean/min/max

If samples are taken in area mode, darktable will calculate mean, minimum and maximum color channel values. This combobox allows you to select which of those are displayed. For obvious statistical reasons mean, min and max are identical for the single sample of point mode.

##### color swatch / color values

A color swatch representing the sampled point or area is displayed alongside numerical values. Clicking on the swatch will toggle on/off a much larger swatch for easier viewing.

The global color picker works in display RGB color space, though you may choose (by selecting from the drop-down) to translate these numerical values into another color space. Beware that values in other color spaces are approximated here – depending on the display color profile there may be some deviations from the actual values.

##### live samples

The sampled colors (in either area or point mode) can be stored as live samples by pressing the “add” button. A color swatch and numerical values will be shown for each stored sample. You can change the numerical value (mean, min, max) and color space to display.

Newly created live samples are not locked. If you change your image processing those changes will be reflected in the live samples. This can be used to see how an altered parameter affects different parts of the image. Clicking on a live sample’s color swatch locks it and a lock symbol is displayed. Further image changes will then no longer affect the sample. You can use this to compare two live samples by locking just one of them, thus providing a before and after comparison.

If you hover the mouse over the “delete” button of one of the live sample entries, the selected region for that sample will be highlighted in the image preview.

**display samples on image/vectorscope**

When this checkbox is ticked, live sample locations are visually indicated on the image and the vectorscope view of the [scopes](#) module.

**restrict scope to selection**


When this checkbox is ticked, only the values of the selected area or point are taken into account by the regular and waveform views of the [scopes](#) module. This allows you to see what tonal values are present in the selected area. When using a color picker in a processing module, this option restricts the scope to the picked area from the processing module instead of the global color picker.

**8.3.1.6. guides & overlays**

A number of commonly-used compositional guides can be overlaid on your image while you are editing. These can be enabled either globally (all the time) or locally (when certain modules are active).

Other darkroom functionality also draws colored overlay lines on the image (for example, drawn masks). An option is also provided to change the color of those overlays (see below).

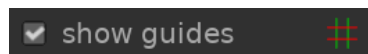
**global guides**

Left-click the  icon in the bottom bar to globally display guide overlays. The overlays will remain switched on until you click the button a second time to switch them off.

Right-click the icon to show the settings dialog (see below).

**local guides**

A more common use is to switch the guides on only when a specific module is activated. The following control is added by default to all modules that crop/distort the image (currently [crop](#) , [crop and rotate](#) , [orientation](#) , [framing](#) , [liquify](#) , [lens correction](#) , [retouch](#) and [rotate and perspective](#) ):



Tick the box to show guide overlays whenever the module is active. Click the icon on the right to show the settings dialog (see below).

**global guide overlay settings**

Please note that, while you can choose to switch guide overlays on and off either globally or locally, the following settings are stored globally and cannot be set independently for each module.

**type**

The type of compositional guide lines to display.

**flip**

Some guides are asymmetrical. This option allows you to flip such guides horizontally/vertically.

**horizontal lines, vertical lines, subdivisions**

When the *grid* overlay type is selected, set the parameters of the grid.

**overlay color**

The color of the overlay lines. Note that this impacts *any* lines that are drawn directly over the image, for example, drawn masks.

**8.3.1.7. history stack**

View and modify the [history stack](#) of the current darkroom image.

This module lists every change of state (activate/de-activate/move/change parameters) for all processing modules that have been modified for the current image. Select a point in the stack to return to that point in the development history of the image. Shift+click an item in the history stack to expand that module in the right-hand module panel without changing the current edit.

**Caution:** If you select a module in the history stack and then make further modifications to the image, all edits above the currently selected step will be discarded. It is easy to lose development work on an image in this way! You can usually use Ctrl+Z to undo such changes.

It is safe to quit the program, leave the darkroom mode, or switch to another image after having selected some earlier state in the history stack module. When returning to that image you will find the history stack panel in the state where you left it.

Hover the mouse over an item in the history stack to show a tooltip with details of all changes that were made in that module compared to its previous or default state. This can help you to track down adjustments that were made unintentionally.

Click “compress history stack” to generate the shortest history stack that reproduces the current image. This causes all edits above the currently-selected step to be discarded. If any module appears multiple times in the remainder of the stack, these will be *compressed* into a single step in the history.

Click “compress history stack” while holding the Ctrl key to truncate the history stack without compressing it i.e. discard all modules above the currently selected one but leaving the remainder of the history stack unchanged.

Click the “reset parameters” button in the module header to discard the entire history stack and reactivate any default modules. This can also be achieved by selecting the “original image” history item and clicking “compress history stack”.

The button to the right of the “compress history stack” button allows you to create a new style from the history stack of the current image, which can then be applied to other images. Use the first line of the popup dialog window to name your style and the second to add a searchable description. You will be prompted to choose which of the modules from the current history stack to include in the style.

Once created, styles can be managed and applied to other images with the lighttable’s [styles](#) module. You can also assign shortcut keys to your styles (see [preferences > shortcuts](#) ) and apply the associated style to all selected images by pressing the shortcut key whenever you are in the lighttable or darkroom view.

### 8.3.1.8. image information line

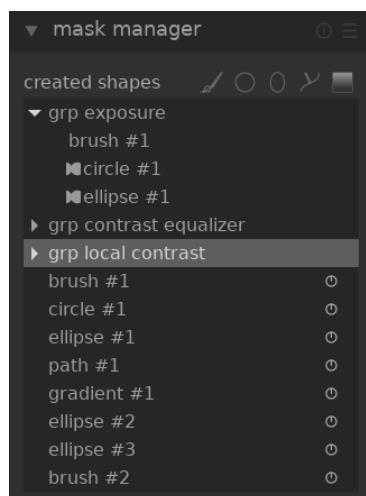
Display information about the current image in a darkroom panel.

The contents of this line can be set in [preferences > darkroom](#) . See the [variables](#) section for information about the variables that can be used here. You can also insert a newline with \$(NL).

The image information line can be displayed in the top, bottom, left or right panel depending on an additional configuration item in [preferences > darkroom](#) .

### 8.3.1.9. mask manager

Manage all masks and shapes for the current image.



This module can be used to create, rename, edit and delete shapes. You can add shapes to and remove shapes from a mask, group shapes together, and combine them using set operators.

The top line of the mask manager panel contains buttons that can be used to create new shapes. These are the same as in the [drawn mask](#) interface of in processing modules.

The panel below these buttons displays a list of all masks and individual shapes defined for the current image.

Groups of shapes forming a mask are displayed with a headline in the form “grp <module\_name>” indicating the module in which they are used, with the component shapes listed below. The list of mask groups is followed by a list of all individual shapes that have been generated in the context of the current image. If a shape is in use by any masks this is indicated by a symbol to the right of the shape name.

## shapes

By default each shape receives an automatically generated name, consisting of the shape type (“brush”, “circle”, “ellipse”, “path”, “gradient”) and an automatically-incremented integer. You can rename a shape by double-clicking on its current name. It is a good habit to give shapes and groups meaningful names, especially if you intend to re-use the same selection in different masks.

Click on a shape name to show the selected shape on the image canvas with all of its controls, allowing you to edit the properties of just that shape. This is especially useful where there are many overlapping shapes within a mask, making it difficult to select the right one with the mouse. Similarly if you select a shape on-screen from within the mask controls of a processing module, that shape will be selected in the mask manager.

Right-click on a shape name to show a menu containing options to remove the current shape or to remove all shapes not currently in use.

---

**Note:** darktable retains all shapes that have ever been defined for the current image unless you explicitly remove them. If you choose to include develop history when exporting an image, all defined shapes will be exported with the image. Beware that if the list of shapes is very long the space required to store those shapes might exceed the size limit of certain file formats. This can cause XMP tag creation to fail during export.

---

## masks & groups

Masks are constructed by adding a group of shapes to the image in the order that they are listed (from top to bottom). Each shape adjusts the existing mask using one of four logical set operators (see below). Because order is important it is also possible to move shapes up and down the list.

Click on the name of a group in the mask manager to expand that group, showing a list of its constituent shapes. The corresponding shapes will be shown on the center image. Similarly if you choose to show a mask from within a processing module, the corresponding group will be expanded within the mask manager.

Right-click on the group name to display a menu with options to add new or existing shapes to the group, or to clean up unused shapes. You can also choose to delete the group.

Right-click on any of the constituent shapes to control how that shape contributes to the overall group mask:

### **remove from group**

Remove the shape from the current mask.

### **use inverted shape**

Invert the polarity of the selected shape.

### **mode**

Choose how this shape will interact with the preceding mask by selecting one of the four set operators defined below.

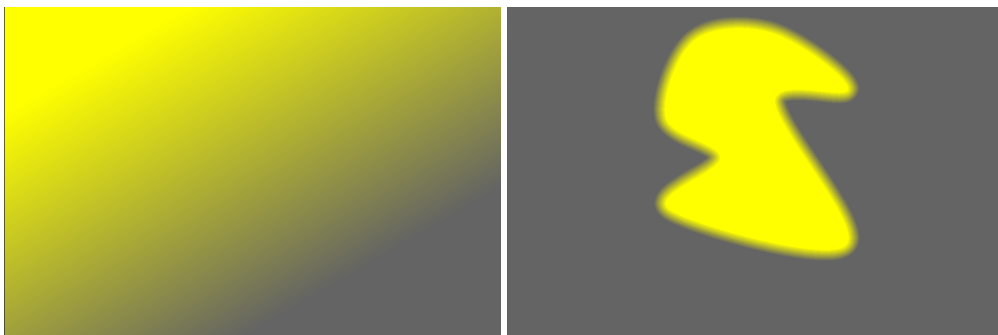
### **move up/down**

Move the shape up or down the list.

You can also create your own groups using existing shapes by selecting the shapes you wish to group, right-clicking them and choosing “group the forms”.


## set operators

Set operators are used to define how grouped shapes are combined. In the following examples we will use a mask that combines a gradient followed by a path, to demonstrate the effect of each set operator when applied to the path shape.




As a convention we say that a pixel is “selected” in a mask or shape if it has an opacity greater than zero.

### union

This is the default set operator and is depicted by the  symbol to the left of the shape name. The shape adds to the existing mask in such a way that the resulting mask contains the pixels that are *either* selected in the existing mask *or* in the added shape. In overlapping areas the maximum value is taken:




### intersection

This set operator is depicted by the  symbol to the left of the shape name. The shape adds to the existing mask in such a way that the resulting mask contains only pixels that are selected in *both* the existing mask *and* the added shape. In overlapping areas the minimum value is used. In the given example we use this operator to “imprint” the path with a gradient:




### difference

This set operator is depicted by the  symbol to the left of the shape name. In the non-overlapping area the existing mask is unchanged. In the resulting mask, pixels are selected only if they are selected in the existing mask but *not* in the added shape. This set operator can be chosen if you want to “cut out” a region from within an existing selection:



### exclusion

This set operator is depicted by the  symbol to the left of the shape name. The resulting mask selects all pixels that are selected in the existing mask and not in the added shape or vice versa. This is equivalent to an “exclusive or”:



### 8.3.1.10. module order

Change the order of the [processing modules](#) in the darkroom using presets.

When processing an image, the active modules are applied in a specific order, which is shown in the right-hand panel of the darkroom view. This module provides information about the current ordering of the processing modules in the [pixelpipe](#) . The name of the currently-selected preset is shown in the module header (or “custom” if the user has manually modified the order). The following presets are available for selection.

#### v3.0 RAW

This is the default module order for [scene-referred](#) RAW development workflow.

#### v3.0 JPEG

This is the default module order for JPEG development.

#### legacy

This module order is used for the legacy [display-referred](#) workflow. You will also see this order displayed for any images you previously edited in darktable prior to version 3.0.

---

**Note:** changing the order of modules in the pixelpipe is not a cosmetic change to the presentation of the GUI – it has real consequences to how the image will be processed. Please do not change the order of the modules unless you have a specific technical reason and understand why this is required from an image processing perspective.

---

For full details of the default legacy and v3.0 module orders, please refer to the [default module order](#) section. For details about changing module order please refer to [the pixelpipe & module order](#) .

### 8.3.1.11. navigation

Zoom and pan the current image.


On the top left hand side of the darkroom, the navigation panel displays a full preview of the current image with a rectangle showing the area that is currently visible in the central panel. Drag the rectangle around to pan the zoomed-in view. The current zoom scale is displayed to the right of the preview image. Click on that figure for a quick access to some common zoom levels.

The navigation panel can be toggled on/off with a keyboard shortcut (Ctrl+Shift+N by default).

### 8.3.1.12. raw overexposed warning

Highlight areas of the image where color channels of the raw input file are clipped.

Clipped color channels imply an overexposed image with loss of information in the affected areas. Some of this information may be recoverable using the [highlight reconstruction](#) , [color reconstruction](#) or [filmic rgb](#) modules.

Click on the  icon to show/hide the warning overlay. Right-click on the icon to open a dialog containing the following configuration parameters.

#### mode

Choose how to mark clipped areas:

- *mark with CFA color*: Display a pattern of the respective primary colors (red, green, and blue) to indicate which color channels are clipped.
- *mark with solid color*: Mark clipped areas with a user defined solid color (see below) independent of the affected color channels.
- *false color*: Set clipped color channels to zero in the affected areas.

#### color scheme

Choose the solid color for the *mark with solid color* mode.

#### clipping threshold

Set the threshold defining what values are considered to be overexposed. You can safely leave this slider at its default value of 1.0 (white level) in most cases.

### 8.3.1.13. snapshots

Store development snapshots and compare with the current edit.

A snapshot is a stored bitmap of the center image in the darkroom view. Snapshots can be taken at any point in the development process and later overlaid onto the current center view. This allows you to undertake a side by side comparison (by default left=snapshot, right=active edit) while you are tuning parameters of a module. This can also be combined with the [history stack](#) module to compare a snapshot against different stages of development.

To take a snapshot, click on the *take snapshot* button. Above the button you will see a list of the snapshots that have been taken for this editing session. The name of each snapshot reflects the name of the module selected in the history stack and its position at the time the snapshot was taken.

Click on the name of a snapshot to show it - this enables a split view between the saved snapshot image and the current state of the processed image. You can control the split view by moving the splitline back and forth over the image with your mouse. If you hover over the splitline with your mouse, a small rotation icon will appear on the center of the line. Click this icon to change between vertical and horizontal split view.

At all times, an arrow containing the letter "S" is displayed to indicate which side of the image is the snapshot and which is the current edit.

Click on the name of the snapshot again to disable the overlay and return to your editing session. Click the module's reset button to remove all existing snapshots.

---


**Note:** Snapshots are retained for the duration of your darktable session. This means that you can also use snapshots to compare with a duplicate edit of the same image. Just navigate to that image and enable the snapshot view as normal.

---



### 8.3.1.14. soft proof

View your image rendered using a selected color profile.

Click the  icon to activate the soft proof display mode on your image. This allows you to preview your image rendered using a printer profile to see how colors will end up on the final print. You can also activate soft proof with the keyboard shortcut Ctrl+S. A message “soft proof” on the bottom left of your image tells you that you are in soft proof display mode.

Right-click on the icon to open a dialog with the following configuration parameters. For each of these parameters, the list of available profiles is read from `$DARKTABLE/share/darktable/color/out` and `$HOME/.config/darktable/color/out` (where `$DARKTABLE` represents darktable’s installation directory and `$HOME` your home directory).

#### **display profile**

Set the color profile for the display. The option “system display profile” is the preferred setting when working with a calibrated display. The profile is taken either from your system’s color manager or from your X display server. The method darktable uses to detect your system display profile can be changed in [preferences > miscellaneous](#) . For more information see the [display profile](#) section.

#### **preview display profile**

Set the color profile for the preview image if you are using a second window.

#### **intent**

Set the rendering intent for your display – only available if rendering with LittleCMS2 is activated. See [rendering intent](#) for a list of available options. This option appears twice – once for the “display profile” and once for the “preview display profile”.

#### **softproof profile**

Set the color profile for soft proofing. Typically these profiles are supplied by your printer or generated during printer profiling.

#### **histogram profile**

Set the color profile of the histogram. None of the available options are ideal, however, “system display profile” is probably the *least bad* setting, since all other profiles are derived from the display color space and at least the values will conform to what you see on screen.

## 8.3.2. lighttable

### 8.3.2.1. history stack

Manipulate the history stack of one or more selected images.

#### module controls

#### **selective copy...**

Copy parts of the history stack from the selected image. A dialog appears from which you may choose the items to copy from the history stack. If more than one image is selected, the history stack is taken from the image that was selected first. Double-click on a history item to copy that item only and immediately close the dialog.

**copy**

Copy the complete history stack from the selected image. If more than one image is selected, the history stack is taken from the image that was selected first.

Information relating to internal display encoding and mask management is considered unsafe to automatically copy to other images and will therefore not be copied when using this button.

The following modules are excluded from the *copy* operation:

- [orientation](#)
- [lens correction](#)
- [raw black/white point](#)
- [rotate pixels](#)
- [scale pixels](#)
- [white balance](#)
- deprecated modules

You can override all of these exclusions by using “selective paste...” and choosing which modules to paste to the target image(s).

**compress history**

Compress the history stack of the selected image. If any module appears multiple times in the history stack, these occurrences will be *compressed* into a single step in the history. *Beware: this action can not be undone!*

**discard history**

Physically delete the history stack of the selected images. *Beware: this action can not be undone!*

**selective paste...**

Paste parts of a copied history stack onto all selected images. A dialog appears from which you may choose the items to paste from the source history stack.

**paste**

Paste all items of a copied history stack onto all selected images.

**mode**

This setting defines how the paste actions behave when applied to an image that already has a history stack. In simple terms the “*overwrite*” mode deletes the previous history stack before pasting, whereas “*append*” concatenates the two history stacks together.

A copied history stack can have multiple entries of the same module (with the same name or different names) and pasting behaves differently for these entries in *append* and *overwrite* modes.

In *append* mode, for each module in the copied history stack, if there is a module in the destination image with the same name it will be replaced. If there is no such module, a new instance will be created. In both cases the pasted instance is placed on top of the history stack. If a particular module appears multiple times in either history stack only the last occurrence of that module will be processed.

In *overwrite* mode the behavior is the same except that the history of the destination image is deleted before the paste operation commences. The “*copy all*”/“*paste all*” actions in this mode will precisely duplicate the copied history stack to the destination images (including any duplicate occurrences).

---

**Notes**

- Automatic module presets are only added to an image when it is first opened in the darkroom or its history stack is discarded. If you use *overwrite* mode to paste history stack entries to images that haven't previously been opened in the darkroom then the next time that image is opened in the darkroom, automatic presets will be applied to the image. It may therefore seem as if the “*overwrite*” mode did not accurately duplicate the existing history stack, but in this case, those automatic modules were added subsequently.
  - The *append* mode allows you to later reconstruct your pre-existing history stack (because previous history items are retained in the stack of the destination image). However, in “*overwrite*” mode all previous edits are irrevocably lost.
  - The *mode* setting is retained when you quit darktable – if you change it for a one-off copy and paste, make sure to change it back again.
-

### load sidecar file

Open a dialog box which allows you to import the history stack from a selected XMP file. This copied history stack can then be pasted onto one or more images.

Images that were exported by darktable typically contain the full history stack if the file format supports embedded metadata (see the [export](#) module for details of this feature and its limitations). You can load an exported image as a sidecar file in the same way as you can with an XMP file. This feature allows you to recover all parameter settings if you have accidentally lost or overwritten the XMP file. All you need is the source image, typically a raw, and the exported file.

### write sidecar files

Write XMP sidecar files for all selected images. The filename is generated by appending “.xmp” to the name of the underlying input file.

By default darktable generates and updates sidecar files automatically whenever you work on an image and change the history stack. You can disable automatic sidecar file generation in [preferences > storage](#) . This can be useful when you are running multiple versions of darktable (so that edits in each version do not conflict with one another) however, in general, disabling this feature is not recommended.

## 8.3.2.2. import

Add images to the darktable library, optionally copying them from another location on the filesystem or from a connected camera.

See [supported file formats](#) for more information.

### module controls

The following buttons are shown in the module’s UI by default:

#### [add to library](#)

Add existing images to the darktable library without copying or moving files. If you only add a single image to the library it will be automatically loaded in the darkroom.

#### [copy & import](#)

Create copies of images from the filesystem and then add those copies to the darktable library.

When a camera is detected, a new section will appear in the module for that device. If you hover your mouse over the camera label, a tooltip will display information about the camera (model, firmware version etc.)

Depending on the capabilities of the camera, the following additional buttons may be displayed:

#### **mount camera**

Mount the camera for exclusive use by darktable. This button only appears if the camera is not currently mounted and is not locked by another process.

#### [copy & import from camera](#)

Create copies of images from the connected camera and then add those images to the darktable library. This button only appears if the camera is currently mounted.

#### [tethered shoot](#)

Open the tethering view so that you can take images with your connected camera using darktable. This button only appears if the camera is currently mounted.

#### **unmount camera**

Unmount the camera and release it for use by other applications. This button only appears if the camera is currently mounted.

### module parameters

Click on the “parameters” label or the expander button beside it to display the following additional options.

All parameters are retained from one session to the next and can be saved as module presets.

#### **ignore exif rating**

Check this option to ignore any rating stored within the image’s Exif data and instead use a hard-coded value (below).

#### **initial rating**

Initial star rating (from 0 to 5) to use for all newly-imported images (default 1).

## apply metadata

Check this option to automatically set metadata fields and/or tags at import time (see below).

## metadata

If the “apply metadata” box is checked, a list of the *visible* metadata fields is presented for completion (see [metadata editor](#) for details). Any populated strings are automatically added to the imported images. You can also choose from any presets saved within the metadata editor module.

Double-click on a label to reset the corresponding field. Double-click on the “metadata presets” label to reset all fields.

When [preferences > storage > write sidecar file for each image](#) is unchecked an additional column “from xmp” is presented so that you can choose to prevent the import of metadata from XMP files.

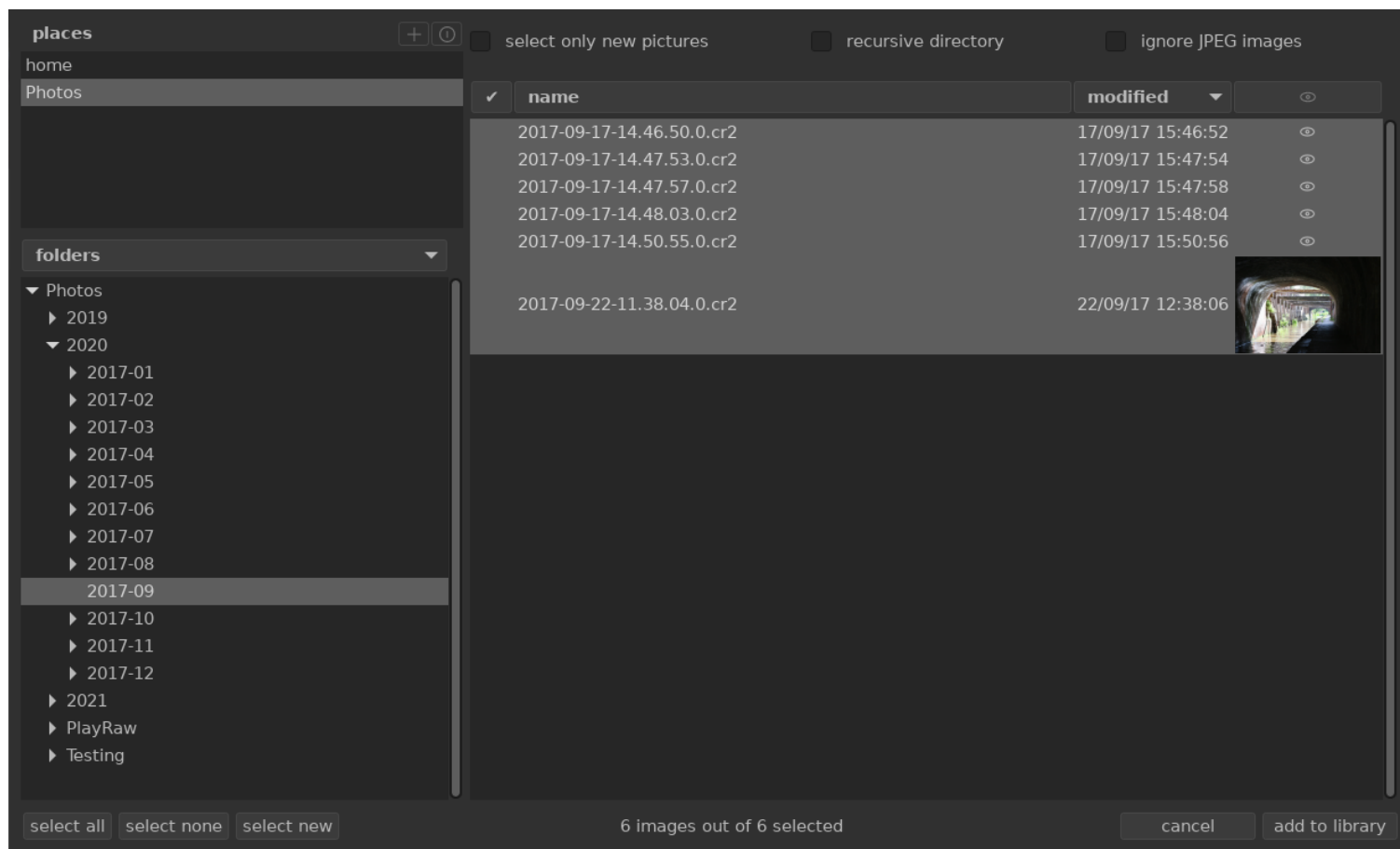
## tags

If you want to add further tags by default when importing images, you can provide them here as a comma separated list. As with metadata you can also choose from any presets saved within the [tagging](#) module.

## import dialog

Each of the three import buttons (add to library, copy & import, copy & import from camera) uses a similar dialog for the import process, described in this section.

The following example screenshot is taken from the “add to library” button:



## common functionality

### places and folders


The import dialog is intended to allow you to set up common import locations, to make subsequent imports as simple as possible. When you first open the dialog, darktable attempts to add some common locations (home, pictures, mounted devices) to the places pane. You can add new places to the list by clicking on the + button and you can remove places from the list by right-clicking on them. If you wish to restore a default location that you have deleted, you can do this with the reset button.

When you choose a place, the folder tree is automatically populated (into the folders pane) from the root directory of the selected place. You can then navigate the folder tree and select a folder for import. The last selected place/folder is automatically reloaded the next time you open the dialog.

In the example screenshot above, you can see that a “place” has been created for the root of the Pictures folder and a sub-folder within that structure has been selected. This is the recommended workflow for the import process – you should not have to create new places very often.

### files

Once you have selected a folder, the files pane will automatically be populated with a list of the files found in that folder. By default, all files in the chosen folder are selected.

You can view thumbnails for any of the images by clicking on the  icon. In addition, buttons are available at the bottom of the screen to select “all” files or “none”.

Once you are happy with your selection, press Enter or click on the button at the bottom-right of the screen to import (this button will be named differently depending on the type of import being performed).

Press Escape or click the “cancel” button to exit without importing.

### common options

The following additional options are common to all import dialogs:

#### **recursive directory**

Check this option to import images in the selected folder *and all subfolders*. It is recommended that you not use this option to import a large number of images at the same time. The import process causes darktable to generate thumbnails for all of the imported images, but in the end it will only be able to keep the most recent in its cache. It is therefore better to import images in smaller chunks to avoid the performance penalty this imposes.

#### **ignore JPEG images**

Check this option if there are JPEG images in the same folder that you do not wish to import. This option is usually used where the camera stores RAW+JPEG and you only want to work on the RAW files, leaving the JPEG images untouched.

### add to library

The “add to library” button allows you to add one or more existing images to the darktable library from the local filesystem. This process does not copy or move images but merely adds their details to the library database and creates XMP sidecar files for them.

#### **select only new pictures**

Tick this box to restrict the *initial* selection (when the dialog is loaded) to only those images that have not already been loaded into the darktable library. If you attempt to reload existing images into the darktable library, data for those images will be reloaded from the XMP sidecar files. A button is also available at the bottom of the dialog to select only “new” images for the currently-selected folder.

---

**Note:** “Add to library” does not create duplicates of your image files in a separate folder structure but processes them in-situ. The “add to library” process simply adds details of those images to darktable’s library database (and creates an XMP sidecar file if applicable) allowing the images to be viewed and developed.

This means that if you delete images from disk after having added them, darktable cannot access them any more. Moreover, darktable does not watch for changes in the filesystem. Any new images will not be shown until they are explicitly imported.

---

## copy & import

This option copies images from another location on your filesystem (including mounted storage devices) and then adds the copied images to the darktable library.

The following additional options are available to control the file and directory naming of the copied files. By default, only the “import job” option is shown – click on the “naming rules” label or the expander icon beside it to show additional options:

### import job

The name of the import job (populated into the \$(JOBCODE) variable).

### override today's date

Enter a valid date/time (in YYYY-MM-DD[Thh:mm:ss] format) if you want to override the current date/time used when expanding the variables \$(YEAR), \$(MONTH), \$(DAY), \$(HOUR), \$(MINUTE) and \$(SECONDS). Leave the field empty otherwise.

### base directory naming pattern

The base directory part of the naming pattern (default \$(PICTURES\_FOLDER)/Darktable). Click on the icon beside the input field to choose a directory manually.

### sub directory naming pattern

The sub directory part of the naming pattern (default \$(YEAR)\$(MONTH)\$(DAY)\_\$(JOBCODE)).

### keep original filename

Check this box to keep original the filename instead of using the pattern below when importing.

### file naming pattern

The file name part of the naming pattern (default \$(YEAR)\$(MONTH)\$(DAY)\_\$(SEQUENCE) . \$(FILE\_EXTENSION)).

### keep this window open

Keep the window open after the import is complete, allowing multiple imports but with different naming options.

Most of these options can also be set in [preferences > import](#) . See this section for more information about the available variables.

## copy & import from camera

This option copies files from a connected camera to the local filesystem and then adds the copied images to the darktable library. It provides the same naming options as the “copy & import” dialog but does not allow places or folders to be selected.

### 8.3.2.3. lua scripts installer

This module provides an interface for installing darktable [lua scripts](#) . The first time it is run, instructions are displayed in the module.

It can be disabled with an option in [preferences > lua options](#) .

### 8.3.2.4. select

Select images in the lighttable according to simple criteria.

## module controls

### select all

Select all images in the current collection.

### select none

De-select all images.

### invert selection

Select all images in the current collection that are not currently selected.

### select film roll

Select all images in the current collection that are in the same film roll as the currently-selected images.

### select untouched

Select all images in the current collection that have not yet been developed.

### 8.3.2.5. selected images

Perform actions on images that have been selected in the lighttable view.

#### module controls

The module controls are spread over two tabs for manipulating the image files and the related metadata.

#### images tab

##### **remove**

Remove the selected images from the darktable library without deleting them. Removed images can no longer be viewed or edited within darktable but the image files remain on the filesystem along with any XMP sidecar files. As darktable keeps the XMP files up-to-date with your latest development history, you can fully restore your work later by re-importing the images.

##### **delete / delete (trash)**

Remove the selected images from the darktable library and remove any associated XMP sidecar files from the filesystem. If no duplicates of a removed image remain in the darktable library, the image file itself is also deleted. You can control whether this action irrevocably deletes the files or attempts to put them into your system's trash bin with a configuration item in [preferences > security](#). A second configuration item in the same tab allows you to control whether or not to be prompted before deleting images.

##### **move**

Physically move selected images (the image files plus all associated XMP sidecars) to another folder on the filesystem. If an image with the given filename already exists in the target folder the source image will not be moved.

##### **copy**

Physically copy selected images (the image file plus all associated XMP sidecars) to another folder on the filesystem. If an image with the given filename already exists in the target folder it will not be overwritten - instead a new duplicate will be generated with the same history stack as the source image.

##### **create hdr**

Create a high dynamic range image from the selected images, and add the result to the library as a new image in DNG format. Images need to be properly aligned, which implies that they have been taken on a sturdy tripod. You can also generate HDRs with programs like Luminance HDR, and later import them into darktable for further processing. Note that darktable can only create HDR images from raw files.

##### **duplicate**

Create duplicates of the selected images within darktable. Duplicate images share the same image file, but each duplicate has its own XMP sidecar file and a separate entry in darktable's library database. This allows you to test different edits on the same image.

##### **rotation**

Perform a clockwise or counter-clockwise rotation on the selected images. A third button can be used to reset the image rotation to the value stored in the image's Exif data. This feature is directly linked to the [orientation](#) processing module - adjustments made here are automatically converted into a history stack item for that module.

##### **copy locally**

Create local copies of the selected images on the local drive. These copies will then be used when the original images are not accessible (see [local copies](#)).

##### **resync local copy**

Synchronize the XMP sidecar of the local copy of each selected image with the copy in external storage, and remove the local copies. Note that if a local copy has been modified and the external storage is not accessible the local copy will not be deleted (see [local copies](#)).

##### **group**

Create a new group from the selected images (see [image grouping](#)).

##### **ungroup**

Remove the selected images from their group (see [image grouping](#)).

#### metadata tab

##### **metadata type checkboxes**

Choose the types of metadata (ratings, tags, metadata, colors, geo tags) that you want to operate on.

##### **copy**

Copy the chosen types of metadata from the selected image onto the clipboard. If you have more than one image selected, or no images selected, then this button is unavailable.



**paste**

Paste any metadata in the clipboard onto the selected images.

**clear**

Clear the chosen types of metadata from the selected images.

**mode**

When pasting metadata onto images, this option controls whether the metadata on the clipboard should be merged with the existing metadata (*merge*), or should replace it entirely (*overwrite*).

**refresh exif**

Refresh the Exif data from the source file. Warning: this may overwrite some tags and metadata that you have altered in darktable (such as star ratings).

**monochrome**

Flag the image as monochrome, meaning that it will receive any monochrome-specific workflow treatment that is offered by the processing modules. Refer to the [developing monochrome images](#) section for more details.

**color**

Remove the monochrome flag from the image so that it will receive the default workflow treatment that is normally used when developing color photos.

### 8.3.2.6. styles

Create named styles from selected images' [history stacks](#) and apply styles to selected images.

Styles can either be created within this panel or in the [history stack](#) module in the darkroom.

A list of all available styles is displayed in the module. A search field above the list allows you to locate a style by name or description. This module also allows styles to be edited and deleted.

Double-click on a style name to apply that style to all selected images. A style may also be applied to all selected images by pressing a shortcut key assigned to it (see [preferences > shortcuts](#) ) while in the lighttable or darkroom view.

Remove all styles by clicking on the module's reset button.

Please note that styles also include the active state of each module. You can use this to create your own default settings, which you can then activate on-demand. Simply set your desired defaults for each module, disable the module, and save the style.

#### module controls

**create duplicate**

When applying a style to images, tick this box to create a duplicate of each image before applying the chosen style to that duplicate. Disable this option to apply the chosen style directly to the selected images. Beware that in this case any existing history stack is overwritten (depending on the mode - see below) and cannot be recovered.

**mode**

As with the [history stack](#) module, this combobox allows you to either "append" the style to the current history stack or to "overwrite" the history stack of the target image.

**create**

Create new style(s) using the history stack of the selected image(s). For each selected image a style creation window is shown. You must supply a unique name for each new style and you can also add an optional description. You will be given the option to de-select any history stack items that you don't want to be part of the created style.

The panel supports a hierarchical view, meaning that you can create categories using the pipe symbol "|" as a separator. For example "print|tone curve +0.5 EV" will create a "print" category with a style of "tone curve +0.5 EV" below it.

**edit**

Select "edit" to bring up a dialog which allows you to include or exclude specific items from the stack of an existing style. Check the "duplicate" option if you want to create a new style, instead of overwriting the existing one, in which case you will need to provide a unique name for the new style.

**remove**

Delete the selected style, without any further prompt.

**import**

Import a previously saved style. Styles are stored as XML files with the extension `.dtstyle`. If a style with the same name already exists, you will be asked if you wish to overwrite.

**export**

Save the selected style to disk as a `.dtstyle` file. This allows styles to be published and shared with other users.

**8.3.2.7. timeline**

View your images by the date/time they were taken.



In the lighttable filemanager view, you can show/hide the timeline module in the bottom panel with the shortcut `Ctrl+F`.

Within the timeline, you can show the next and previous dates by scrolling your mouse; `Ctrl+scroll` to zoom in/out.

You can also use the timeline to select images by date range by clicking and dragging your mouse.

**8.3.3. map****8.3.3.1. find location**

Search for a location on the map. You must be connected to the internet to use this feature.

To use this module, type in a place name or address, press `Enter` and a list of results will be shown. Click on an item in the list and the map will zoom to that location. An outline covering that location or a pin pointing at the location will be displayed.

An outline (polygon) can be used to create a user location. Check the *max polygon points* parameter in the [map settings](#) module to ensure that sufficient points are available for a polygon to be displayed.

**8.3.3.2. locations**

Create areas or locations and organize them as hierarchical tags.

A location is shown as a shape on the map when selected. Initially each location is represented as a square or circle and can be changed to a rectangle or ellipse by adjusting the shape's width and/or height.

A location can also be created from an OpenStreetMap region (city/country) polygon. To achieve this, first make sure the *max polygon points* parameter is large enough (some country polygons use more than 150,000 points). Then select the desired location in the [find location](#) module. When the corresponding region shape is displayed, a polygon symbol becomes available in the "shape" control (see below). Select it to create the new location.

Each location is stored as tag entry under the geotagging collection in the [collections](#) module. The pipe "|" character can be used to insert a new level (a group of locations).

**module controls****shape**

Select the circle or rectangle symbol to choose the default shape for new locations. A polygon symbol is available when a shape is displayed by the [find location](#) module.

**new location / new sub-location**

When no location is selected you can use the *new location* button to create a location at root level. When a location is selected use the *new sub-location* button to create a sub-location within the selected location.

**show all**

Show or hide all locations that lie within the visible area of the current map.

**actions on the locations list****click**

Select or de-select a location. If the location is not currently visible on the map, the map is automatically centered on that location.

**Ctrl+click**

Edit the name of the location. Press Enter to save changes or Esc to close the editing window without saving.

**right-click**

Open a sub-menu which allows you to:

- Edit the name of the location.
- Delete the location.
- Update the filmstrip – the filmstrip will be populated with all images in the selected location.
- Switch to the lighttable view and show a collection that contains all images in the selected location.

actions on locations in the map

*Note: The following actions have no effect on polygon locations.*

**click and drag**

Move the location shape to a new position on the map.

**Ctrl+click or Ctrl+Shift+click**

Move an image or a group of images and place them inside a location shape.

**mouse scroll**

When inside a location shape (but not over an image), increase or decrease the size of that shape.

When hovering over an image, cycle through the thumbnails of images located at that position on the map.

When outside a location shape (and not over an image) zoom the map in or out.

**Shift+scroll**

Increase or decrease the width of the location shape.

**Ctrl+scroll**

Increase or decrease the height of the location shape.

**click on a location shape**

Select a different location when the *show all* checkbox is selected.

### 8.3.3.3. map settings

Select preferred map data from various providers. Some will provide additional layers (satellite view etc.) which you can toggle.

module controls

**map source**

Choose the provider to source map information from.

**max polygon points**

The [find location](#) module doesn't display polygons with more points than this for performance reasons. Usually a country polygon has between 50,000 and 150,000 points.

**show OSD**

Choose whether to display the OSD controls at the top-left of the center view.

**filtered images**

Check this box to display only the images from the current collection (those shown in the filmstrip) in the center view. Uncheck the box to display all images in the current library, where those images have associated GPS data. You can also toggle this option by pressing Ctrl+S.

**max images**

The maximum number of thumbnails to display on the map.

**group size factor**

Increase or decrease the size of area that causes images to be grouped.

**min images per group**

The minimum number of images that need to be placed in the same position in order to automatically create an image group for them.

**thumbnail display**

Define what information to show on the map display

- *thumbnails*: Display image thumbnails along with a counter.
- *count*: Just display the number of images (to free space on the map). Hover over the number of images to show the corresponding thumbnail(s). A count-only marker behaves the same way as a normal image thumbnail, in terms of color coding, scrolling, drag and drop etc.
- *none*: Show nothing.

You can also cycle through these options by pressing Shift+S.

## 8.3.4. print

### 8.3.4.1. print settings

Manage settings for the [print view](#) and initiate printing.

module controls

printer

**printer**

Select one of the installed printers.

**media**

The type of media loaded on the printer (Plain Paper, Luster Photo Paper, etc.).

**profile**

The printer's ICC profile for the loaded paper. This is the profile specific to the printer and paper. This profile is the last color space transformation applied to the picture whose goal is to create a high quality print.

**intent**

The print rendering intent ("perceptual", "relative colorimetric", "saturation" or "absolute colorimetric"). See [rendering intent](#) for more details.

**black point compensation**

Whether to adjust the black point of the output profile, which is often lighter than the input profile. This should be "on" when the *intent* is set to "relative colorimetric".

page

**paper size**

The size of the paper on which to print.

**orientation**

Portrait or landscape (note that darktable chooses the best fit by default).

**units**

The unit to use for setting the margins: "mm", "cm", or "inch".

**margins**

Set each margin separately, or all together by clicking on the middle "lock" button.

**display grid**

Select the grid size using the entry field (expressed in the currently selected unit). Tick the option to display the grid on the canvas.

**snap to grid**

Help setting the image areas by snapping them to the grid for proper alignment.

**borderless mode required**

Indicates whether the printer borderless mode is to be activated. This item is activated when the user's margins are smaller than the printer hardware margins. Note that it is only an indicator as it does not activate the borderless mode automatically.

image layout

**image width/height**

This information field displays the actual image width and height (given with the selected units) on the paper.

**scale factor**

This information field displays the scaling of the image to fit on the paper. If this value is less than 1 the image is down-scaled, otherwise it is up-scaled. This is an important factor to watch – a value that is too large (up-scale) may result in a low quality print. The corresponding dpi (dots per inch) is also displayed.

**alignment**

Select the alignment of the image on its area.

**new image area button**

Create a new image area. Drag and drop on the canvas to place it. If the option snap to grid is activated the area can be easily aligned to the grid lines. An image can be placed into this area by dragging it from the filmstrip and dropping it on the new area.

**delete image area button**

Remove the currently selected image area from the composition.

**clear layout button**

Remove all the image areas leaving the canvas empty.

The following four fields represent the position of currently selected area on the page – the top/left corner on the first line and the width/height of the area on the second line.

When hovering an image area its position and size are displayed. It is also possible to grab the side and corner of the area to change the size or to drag the whole area to change its position.

The page layout can be recorded using a preset.

print settings

**profile**

The export profile to use. This profile is the entry point used for the next transformation using the printer's ICC profile. Usually it is better to prefer a large gamut (e.g. Adobe RGB) rather than a smaller one (e.g. sRGB).

**intent**

The rendering intent to use when exporting the image. For more information see [rendering intent](#) .

**style**

Choose a style to apply when exporting the image – defaults to “none”. See the [export](#) module for a more detailed discussion of applying a style during export.

**mode**

Whether the chosen style should be appended to the existing history stack or replace it completely. See the [export](#) module for more details.

print button

When clicked, the images are first exported using the selected options, then composed on the page and finally sent to the printer.

## 8.3.5. shared

### 8.3.5.1. collections

Filter the images shown in the lighttable view and filmstrip panel using image attributes. This set of filtered images is known as a *collection*.

Importing images into darktable stores information about them (filename, path, Exif data, data from XMP sidecar files etc.) in darktable's library database. A collection may be defined by applying filtering rules to these attributes, thus creating a subset of images to display in the lighttable view and the filmstrip module.

The default collection is based on the *film roll* attribute and displays all images of the last imported or selected film roll.

filtering attributes

The images in a collection can be filtered using the following image attributes:

files

**film roll**

The name of the film roll to which the image belongs (which is the same as the name of the folder in which the image resides). Ctrl+Shift+click on a film roll to switch to the corresponding folder. Right-click to remove the contents of the film roll from the darktable library or tell darktable that its location has changed in the file system.

**folder**

The folder (file path) where the image file is located. Click on a folder to include the contents of that folder and all sub-folders in the collection; Shift+click to include only the images in the selected folder; Ctrl+click to show only the images from any sub-folders; Ctrl+Shift+click to switch to the corresponding film roll. Right-click on a folder name to remove its contents from the darktable library or tell darktable that its location has changed in the file system.

**filename**

The image's filename.

metadata

**tag**

Any tag that is attached to the image. Untagged images are grouped under the "not tagged" entry. When activated, a hierarchical list of known tags is displayed

**title**

The image's metadata "title" field.

**description**

The image's metadata "description" field.

**creator**

The image's metadata "creator" field.

**publisher**

The image's metadata "publisher" field.

**rights**

The image's metadata "rights" field.

**notes**

The image's metadata "notes" field.

**version name**

The image's metadata "version name" field.

**rating**

The image's star rating

**color label**

Any color label attached to the image ("red", "yellow", "green", "blue", "purple").

**geotagging**

The geo location of the image (see [locations](#) ).

times

**date taken**

The date the photo was taken, in the format YYYY:MM:DD.

**date-time taken**

The date & time the photo was taken, in the format YYYY:MM:DD hh:mm:ss.

**import timestamp**

The date/time the file was imported, in the format YYYY:MM:DD hh:mm:ss.

**change timestamp**

The date/time the file was last changed, in the format YYYY:MM:DD hh:mm:ss.

**export timestamp**

The date/time the file was last exported, in the format YYYY:MM:DD hh:mm:ss.

**print timestamp**

The date/time the file was last printed, in the format YYYY:MM:DD hh:mm:ss.

## capture details

### **camera**

The Exif data entry describing the camera make and model.

### **lens**

The description of the lens, as derived from Exif data.

### **aperture**

The aperture, as derived from Exif data.

### **exposure**

The shutter speed, as derived from Exif data.

### **focal length**

The focal length, as derived from Exif data.

### **ISO**

The ISO, as derived from Exif data.

### **aspect ratio**

The aspect ratio of the image, including any cropping within darktable.

## darktable

### **grouping**

Choose between “group followers” and “group leaders”.

### **local copy**

Show files that are, or are not copied locally.

### **history**

Choose images whose history stacks have been altered or not.

### **module**

Filter based on the processing modules that have been applied to the image.

### **module order**

Choose images with “v3.0”, “legacy” or “custom” module orders.

## module controls

## defining filter criteria

The top line of the module can be used to define filter criteria for your collection as follows:

### **image attribute**

The combobox on the left allows you to choose which [attribute](#) to use to filter your collection.

### **search pattern**

In the text field to the right of the attribute combobox, you can write a matching pattern. This pattern is compared against all database entries with the selected attribute. The filtering mechanism detects a match if any image’s attribute contains the pattern in its full text. You may use % as wildcard character. The collection will be limited to only the matching images. Leave the text field empty to match all images having that attribute. Where applicable, a tooltip will appear if you hover over the attribute or search pattern to provide additional information.

Attributes with numerical or date/time attributes can be used in combination with comparative and range operators. Use <, <=, >, >=, <>, or = to select images with attributes less than, less than or equal, greater than, greater than or equal, not equal, or equal to the given value, respectively. An expression in the form [ from; to ] can be used to select using a range of values.



**select by value**


As well as defining filter criteria using a search pattern, you can also manually choose from a list of values (for the chosen attribute) taken from the currently-matching set of images. Making such a selection will automatically populate the “search pattern” field.

The box below the *search pattern* will list values for the selected attribute that are present in the currently-selected images. This list is continuously updated as you type. You may also choose a sorting criteria by scrolling through the list and double-clicking.

If you enable single-click mode (see [preferences > lighttable](#) ) you can select with a single-click rather than double-click. This mode also allows you to select a range of values with the mouse. This only works for numerical and date-time attributes.

**combining multiple filters**

You can combine multiple filters together to create more complex collections of images using a series of rules. A rule is a combination of a filter criteria along with a logical operation that defines how that criteria is combined with any previous rules.

Click on the  button (to the right of the search field) to open a menu with the following options:

**clear this rule**

Remove the current rule, or reset it if this is the only rule defined.

**narrow down search**




Add a new rule, which is combined with the previous rule(s) in a logical AND operation. An image is only retained as part of the collection if it also fulfills the new criteria.

**add more images**

Add a new rule, which is combined with the previous rule(s) in a logical OR operation. Images that fulfill the new criteria are added to the collection.

**exclude images**

Add a new rule, which is combined with the previous rule(s) in a logical EXCEPT operation. Images that fulfill the new criteria are removed from the collection.

The logical operators defining how rules are combined are indicated with icons to the right of each added rule: AND by the  symbol, OR by the  symbol, and EXCEPT by the  symbol. Click on any of these icons to change the logical operation for that rule.

**updating the folder path of moved images**

While it is best to not touch imported files behind darktable’s back, this module can help you to recover from situations when you have moved or renamed image folders after importing them. The collections module has a feature that allows you to update darktable’s library database with the new folder location. The process is as follows:

1. Set the *image attribute* combobox to “folder” or “film roll”.
2. The original film roll or folder name will be displayed with ~~strikethrough~~ formatting to emphasize that it can not be located.
3. Right-click on the folder or film roll name, select “search filmroll...”, and then select the new location of the folder.

**preferences**

The “preferences...” option in the presets menu allows you to adjust the behavior of the collections module as follows:

**do not set the ‘uncategorized’ entry for tags**

Do not set the ‘uncategorized’ category for tags that do not have children (default off).

**tag case sensitivity**

Set whether tags should be case sensitive or not - without the `sqlite ICU` extension this will only apply to the 26 latin letters (default insensitive).

**number of folder levels to show in lists**

The number of folder levels to show in film roll names, starting from the right (default 1).

**sort film rolls by**

Sort film rolls by either the “folder” (path) or the “id” (roughly equivalent to the date the film rolls were first imported) (default “id”).

### sort collections descending

Sort the following collections in descending order: “film roll” (when sorted by folder), “folder”, date/time (e.g. date/time taken) (default on)

### 8.3.5.2. export

Export selected images.

When used in the darkroom view, the currently-edited image will be exported if no other images are selected in the filmstrip.

Files can be exported to a file on disk, email, various online storage locations, a web album, or a book template.

### module controls

### storage options

#### target storage

The type of location to store your selected images. A number of different back-ends are implemented, including file on disk, LaTeX book template and various web albums. Depending on the selected target, you will be asked to provide additional information, such as filenames or account name and password.

#### filename template

Define the folder and file to which the image will be exported. This can be automatically generated using several pre-defined variables. See the [variables](#) section for details.

#### output directory selector

The button beside the filename template entry opens a dialog to select the parent directory for export.

#### on conflict

Choose what to do if the generated filename conflicts with an existing file on export:

- *create unique filename*: Automatically choose a unique new file name by appending an integer to name of the conflicting file.
- *overwrite*: Automatically overwrite existing files. This option will present you with a confirmation dialog in order to protect you from accidental data loss – you can disable this in [preferences > security > ask before exporting in overwrite mode](#). **Note:** *This dialog is not presented per-file but as a one-off confirmation before the export job starts.*
- *skip*: Do not export images where the destination filename already exists.

### format options

#### file format

Choose the file format for the exported image. Additional options will appear (below) depending on the selected format.

#### quality

The quality of the exported file. Higher values lead to larger file sizes. The default quality (95) is a good setting for very high quality exports (e.g. for archiving or printing purposes). If you need a good compromise between size and quality (e.g. for online image display or uploads) you should consider a value of “90” instead.

#### bit depth

The number of bits used for each color channel. More bits means less posterization/color banding.

#### compression

The type of compression to use.

#### compression level

For export formats where *compression* can be specified, the *compression level* specifies how much compression to apply. The higher the level, the more the data will be compressed, at the cost of more CPU cycles.

#### b&w image

For TIFF export format, it is possible to save a monochrome image. This setting controls whether the resulting file encodes the shades of gray as separate RGB channels, or as a single grayscale channel. The latter option will result in smaller files.

## global options

### set size

Choose how to measure the maximum size of your exported image

- *in pixels (for file)*: Enter the maximum width and height in pixels.
- *in cm (for print)*: Enter the maximum width and height in cm and define the image's dpi. The equivalent size in pixels will be automatically calculated.
- *in inch (for print)*: Enter the maximum width and height in inches and define the image's dpi. The equivalent size in pixels will be automatically calculated.
- *by scale (for file)*: Enter a multiplier to specify by how much the exported image should be scaled compared to the input image. For example, entering a value of 0.5 will result in an output image with half the width and height (in pixels) of the original image.

### dpi

If units of cm or inches are chosen, set the dpi of the output image. The dpi will also be stored in the Exif data of the exported image. It will be automatically set to 300 if "in pixels" or "by scale" is chosen.

### max size

Set the maximum width and height of the exported image(s) in pixels, cm or inches (depending on the selected unit) – zero means that no constraint will be set on that dimension. Exported images will be constrained so as not to exceed either of these values, while retaining the correct aspect ratio. Set both to zero to export with the original dimensions (after cropping). If the entered values exceed the original dimensions darktable will either export with the original dimensions or upscale the image, depending on the "allow upscaling" parameter.

### allow upscaling

Set to "yes" to perform an upscaling step if the user-defined maximum width and height exceed the original dimensions of the image. If set to "no" the exported image's dimensions will not exceed the dimensions of the original image (after cropping).

### high quality resampling

Set this to 'yes' to perform high quality resampling on the image. The image will be processed in full resolution and only downsampled at the very end. This can sometimes result in better quality, but will always be slower.

### profile

The output color profile. Select "image settings" if you want the settings in the [output color profile](#) module of the individual images to be respected.

### intent

This option lets you define the intent – the way in which darktable will handle out-of-gamut colors. See [rendering intent](#) for a more detailed description of the available options.

### style

Choose a [style](#) which darktable will combine with the existing history stack to generate the output image. These history items are only added temporarily – the original history stack is not overwritten. You can use this feature to add processing steps and parameters that you want to be applied specifically to images before export. For example you may define a style that adds a stronger level of sharpening when you produce scaled-down JPEG files for the internet or add a certain level of exposure compensation to all of your output images.

### mode

When applying a style during export this option defines whether the history stack items of that style replace the original history stack of the image or are appended to it. Technically speaking, in append mode history stack items of the style will constitute separate instances of the respective modules on top of any existing ones (see also [multiple instances](#)). As a consequence the original history stack will remain in effect with the new items being applied in addition. This way you can apply an overall adjustment (e.g. exposure) to a number of exported images while respecting the settings of each individual image.

### export

Press this button to start a background job to export all selected images. A bar at the bottom of the left hand panel displays the progress of the export job. Furthermore a notification message pops up reporting the completion of each individual export. You may click on the pop-up to make it disappear. You may abort the export job by clicking on the "x" icon located close to the progress bar.

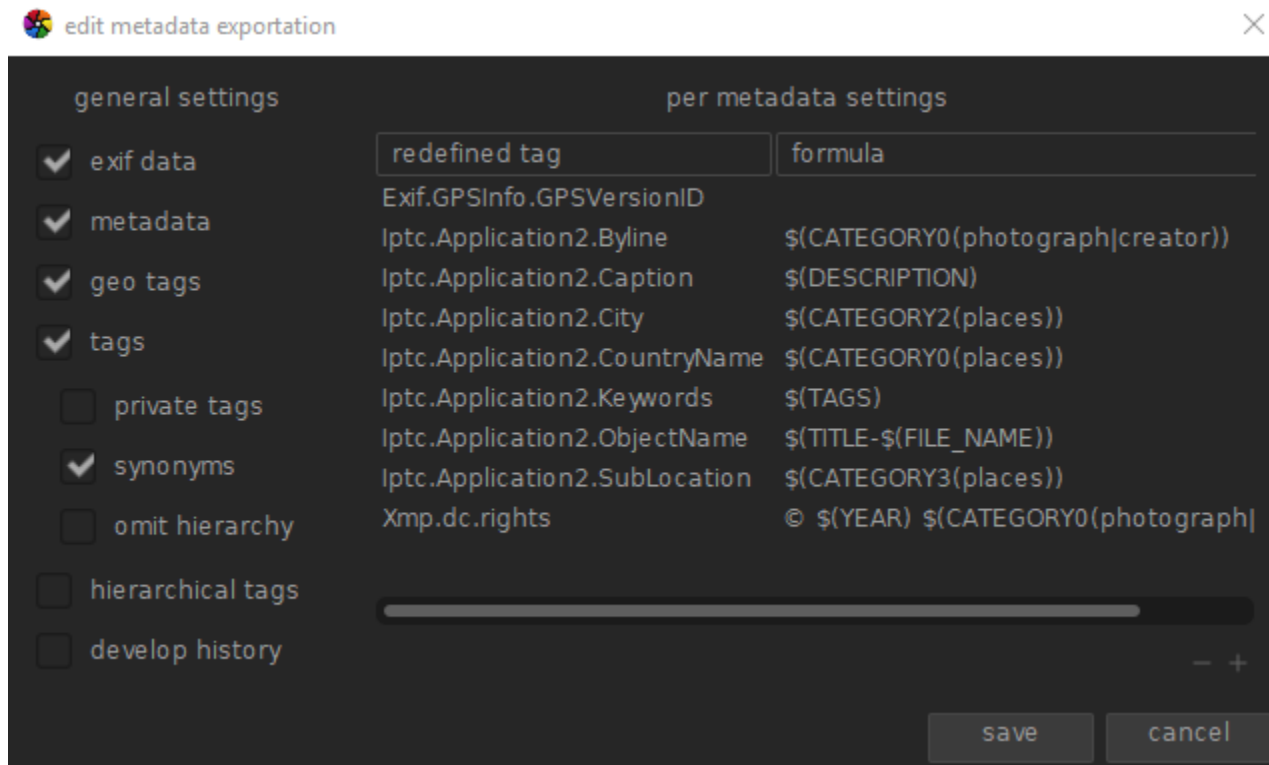
---

**Note:** Images that are selected but currently hidden (because they are members of a collapsed group) will not be exported.

---

## metadata preferences

The “preferences...” option in the presets menu brings up a dialog where you can configure what metadata to include in exported files:



The parameters entered into this dialog are saved along with other export parameters to user presets and the last entered values are retained when darktable is closed.

### general settings

The left-hand-side of this dialog allows you to choose which groups of metadata are to be exported with the image. The following options are available:

#### exif data

Export the source image’s Exif data.

#### metadata

Export metadata defined in the [metadata editor](#) module. Metadata fields tagged as *private* or *hidden* will not be exported.

#### geo tags

Export geo tags.

#### tags

Export tags created in the [tagging](#) module (to Xmp.dc.Subject). Three additional options can also be selected:

- *private tags*: Export private tags
- *synonyms*: Export tag synonyms
- *omit hierarchy*: Only export the last part of hierarchical tags

#### hierarchical tags

Export hierarchical tags (to Xmp.lr.Hierarchical Subject)

#### develop history

Export the entire development history (history stack and shapes) where supported (e.g. JPEG, JPEG2000, TIFF). The development history will be stored as XMP tags within the output file. This information can later be used to reconstruct the parameters and settings that were used to produce the exported image.

**Caution:** For various reasons embedding XMP tags into output files may fail without notice, e.g. if certain size limits are exceeded. Users are therefore advised to not rely on this feature for their backup strategy. To back up your data always make sure to save your input (raw) file as well as all of darktable's XMP sidecar files.

### per metadata settings

The right-hand-side of this dialog allows you to define formulas to populate image metadata. The formulas defined here have priority over the settings in the left-hand-side of the dialog. The first column identifies the entry to be edited. The second column allows you to define how to calculate the value for that metadata entry using a formula.

See the [variables](#) section for details of the variables you can use in your metadata formula. Press Enter to validate the formula. Leave the formula empty to prevent a given metadata entry from being exported (Exif.GPSInfo.GPSVersionID in the above example).

Use the “-” icon to remove a metadata entry from the list and the “+” icon to add a new one from a predefined list of available metadata tags.

Click on the “add” button to add a metadata entry to the list.

The formulas allow you virtually define all the metadata you need to qualify your images in tagging and export the values in the XMP or IPTC tags of your choice. The exported tags can be different from one export to the next depending on the destination of the images. Tags and Categories are displayed separately in image information.

*Remember that a tag set up as a category is never exported.*

### tips

- To prevent a specific metadata field from being exported, add it to the list and leave the formula empty.
- To force a specific exif metadata field to be exported when exif export is disabled, add it to the list and enter = into the formula.

### examples

#### example 1

A first level tag called places is set as a category, and is followed by four levels of information (or keywords): country, region, city and location (e.g. places|France|Nord|Lille|rue Nationale). Each level can be retrieved (when it is defined) by one of the variables \$(CATEGORY0(places)), \$(CATEGORY1(places)), \$(CATEGORY2(places)) and \$(CATEGORY3(places)). In this example, the returned values are “France”, “Nord”, “Lille” and “rue Nationale”, respectively. These keywords can also be retrieved as simple tags using the variable \$(TAGS). The last keyword level defined (the leaf) is displayed in [image information](#), here “rue Nationale”.

#### example 2

A first level tag called creator is followed by the name of the photographer, both set as categories: creator|firstname lastname. The formula copyrights \$(YEAR) \$(CATEGORY0(creator)) builds the text associated with image rights. Here, [image information](#) displays “creator: firstname lastname” as categories. Neither creator nor “firstname lastname” appear in the tags list and they are not exported as simple tags.

**Note:** tagging is not appropriate to define free text metadata, like a title or a description, which may be specific to each image. Use the [metadata editor](#) for this type of information.

### 8.3.5.3. filmstrip

The filmstrip can be used to quickly switch between images in any darktable view. The images shown are the same as those displayed in the lighttable view and are defined by the currently-selected collection.



The filmstrip can be switched on and off using the shortcut Ctrl+F. The height of the filmstrip panel can be changed by clicking and dragging its top border.

Quickly navigate through the images in the filmstrip by scrolling with the mouse. In the darkroom you can change the photo currently being processed by clicking on another image in the filmstrip.

In the darkroom, the image currently being processed is selected and highlighted. Hover over an image on the filmstrip with your mouse to select it (in order to act on it with a keyboard shortcut) without changing the image being processed.

If you wish to select multiple images in the filmstrip, use Alt+click to select the first image, followed by either Ctrl+click to select or de-select further images, or Shift+click to select a range of images.


The following shortcuts are available to select images in the filmstrip or to perform actions on the selected images:

Ctrl+A	select all images in the filmstrip
Ctrl+Shift+A	deselect all images
Ctrl+I	invert the current selection
F1 – F5	toggle color label (red, yellow, green, blue, purple)
0 – 5	set/change image rating
R	reject image
Ctrl+D	duplicate image
Ctrl+C	copy the full history stack
Ctrl+V	paste all of the copied history stack
Ctrl+Shift+C	selectively copy the history stack
Ctrl+Shift+V	selectively paste from the copied history stack

See the lighttable's [history stack](#) module documentation for more information about the copy and paste functionality.

#### 8.3.5.4. focus peaking

Identify which parts of the image contain sharp edges, which usually indicates that those areas are in focus.

Activate the module by clicking on the  icon. The sharp parts of the image will be highlighted with a yellow, green and blue overlay:

Focus peaking works by filtering out image noise, measuring the intensity gradients in the image and calculating average and standard deviation statistics. When the gradient of an edge differs significantly from the mean, the associated pixels are marked with a “heat map” indicating how sharp the edge is.

- *yellow* represents a large ( $6\sigma$ ) jump in gradient, indicating a very sharp edge.
- *green* represents a medium ( $4\sigma$ ) jump in gradient, indicating a reasonably sharp edge.
- *blue* represents a small ( $2\sigma$ ) jump in gradient, indicating a slightly sharp edge.





The image above was taken with a wide aperture to give a shallow depth of field, and you can see how the camera has focused on the chinese character written on the second red lantern along from the front. There are also stems of the pink flowers that fall within the area of acceptable sharpness around the focal plane, and these have also been marked up with yellow and green.

#### 8.3.5.5. geotagging

Import and apply GPX track data to a selection of images.

This module is common to the [lighttable](#) and [map](#) views. The map view provides an enhanced mode that allows you to preview the position of the images along the GPS tracks while adjusting the images' date/time offset and time zone.



## workflow overview

A GPS receiver calculates its current position based on information it receives from satellites, and records it in a GPX file together with the current date and time. The Exif data of the images also contains a time stamp defined by the camera settings. The *geotagging* module takes the time stamp of the image, looks up the position in the GPX file at that time, and stores the appropriate coordinates (latitude/longitude/elevation) in its database and the image's XMP sidecar file.

Two problems may occur during this process:

- In contrast to GPS devices, most cameras don't record the time accurately.
- The time stored in the Exif data does not include the time zone. Most people set their camera to local time, whereas GPS devices store the time in the UTC (Universal Time, Coordinated, i.e. Greenwich (London)) time zone. If the time zones of the camera and GPX file differ, then the related location will be wrong.

If your image already shows the correct date/time and carries the UTC time stamp, you can directly apply the GPX track file without further adjustments.

Otherwise follow the following process to correlate the time of the images and GPS track files

1. Fix the camera time setting for a single image by manually entering the correct date/time for that image into the *date/time* field. A good way to do this accurately is to take a photograph of a reliable time source. This can be any precise clock or, even better, the time displayed on your GPS device (bearing in mind that GPS devices normally *show* the local time, even though they *store* universal time). The difference (offset) between the time entered and the time stored in the image's Exif data will be shown in the *date/time offset* field.
2. Press the *lock* button to lock the calculated offset in the module.
3. Select all of the images you wish to adjust and click *apply offset* to apply the calculated offset to those images.
4. If the camera time zone is not UTC, set the time zone in the *camera time zone* field.

With the time setting corrected, you can now apply GPX tracking data using the *apply GPX track file* button.

## module controls (common)

### date/time

This field is initialized with the date/time read from the first selected image (format yyyy:mm:dd hh:mm:ss) and can be modified to correct the date/time for that image. The individual date/time fields can be altered by scrolling over them with your mouse. If any field reaches its limit, the neighboring fields are automatically updated. For example, if you go over 60 on the minute field, the hour field will automatically be incremented.

### original date/time

The original date/time of the image is shown here for reference.

### date/time offset

The calculated difference between the original date/time and that keyed in the *date/time* field. If the calculated difference is greater than 99 days 23 hours 59 minutes and 59 seconds, the offset is invalidated.

### lock button

If the lock button is activated, the offset value is frozen. If you subsequently change the selected image(s), the new image date/time is updated using the locked offset. This allows you to apply the same offset to multiple groups of images.

### apply offset button

Apply the offset to selected images.

### apply date/time button

It is sometimes useful to be able to set the absolute date/time for an image, for example where this information is missing. This button allows you to apply the date/time entered in the *date/time* field to the selected images, without considering the previous value. You can use Ctrl+Z to undo any unwanted changes.

### camera time zone

Select the camera's time zone. Start typing to show a list of permitted values.

### apply GPX track file (lighttable view only)

Apply a GPX track file. Click the corresponding button and navigate to the GPX file. You can use Ctrl+Z to undo any unwanted changes. Within the file chooser window, the preview button lists the tracks of the selected GPX file along with the following information: track name, start and end date/time (local time), number of track points and number of selected images that will be geotagged.

## module controls (map view)

### gpx file

The path of the gpx file selected.

### track list

This table shows the start date/time of each track, along with the number of track points and the number of matching images. When a check button is activated, the related track is displayed on the map. A check button in the table header allows you to select or de-select all of the tracks at once. Hover over a row with your mouse to display the start and end times both in local time (LT) and UTC.

### preview images

If checked, the matching images are displayed on the map along the visible tracks.

### select images

If you don't wish to apply an offset to all of the selected images, but only to the matching images, use this button to select images. If you don't want to lose the current offset you may want to lock it before changing the selection.

### counts

A count of the number of matching images and the number selected is displayed to the right of the *select images* button.

### apply geo-location

This button is displayed when the offset is null. The *apply geo-location* button applies the GPX data to matching images on selected tracks.

### apply offset and geo-location

This button is displayed once an offset has been entered. As a reminder, the *apply offset* button applies the offset to all selected images. Unlike *apply offset*, the *apply offset and geo-location* button applies the offset *and the GPX data* to matching images on selected tracks.

You can use Ctrl+Z to undo any unwanted changes (twice in case of *apply offset and geo-location*).

## 8.3.5.6. image information

Display information embedded within an image's Exif data as well as a number of additional data fields defined by darktable.

When hovering with the mouse over image thumbnails, the displayed data is automatically updated to show information about the image currently under the mouse cursor.

When several images are selected and the focus is not on a single image, the module only displays information that is the same for all images. If any fields differ between the images, the text "*<various values>*" is displayed instead.

While you are in the lighttable view, you can double-click on the *filmroll* field for a given image to show all images in that image's film roll.

## preferences

The "preferences..." option in the presets menu brings up a dialog with a list of all fields that are available for display.

The *visible* checkbox allows you to choose which fields to display. You can also drag and drop one row at a time to change the display order.

These preferences can be saved as module presets. Press the module's reset button to make all available information visible and displayed in its default order.

## 8.3.5.7. metadata editor

Edit the metadata of selected images.

Metadata is freeformat text (title, description, creator, publisher, rights etc.) that describes your images.

## module controls

### metadata entry fields

A separate field is displayed for each metadata item. Hold Ctrl while scrolling with your mouse to increase the height of a field. Press Ctrl+Enter to insert a new line. Press Enter to apply any text entered in the current metadata field to the selected images. Double-click on a field's label to delete the contents of that field.

**apply**

Apply new settings from the metadata entry fields to the selected images.

**reset button**

Delete existing metadata from the selected images.

## preferences

The “preferences...” option in the presets menu brings up a dialog where you can configure how metadata is handled within darktable. For each metadata item, two check boxes allow you to restrict how metadata is handled:

**visible**

Show or hide this metadata field. Hidden fields are not displayed anywhere in darktable and are not be included in exported images.

**private**

Keep this metadata field private. Private fields are displayed within darktable but are not be included in exported images.

Fields with both options unchecked are displayed within darktable and included in exported images.

### 8.3.5.8. recently used collections

Display a list of recently used collections generated by the [collections](#) module.

Click on an entry to reopen the selected collection. This also updates the collections module with the appropriate filter criteria and rules.

## preferences

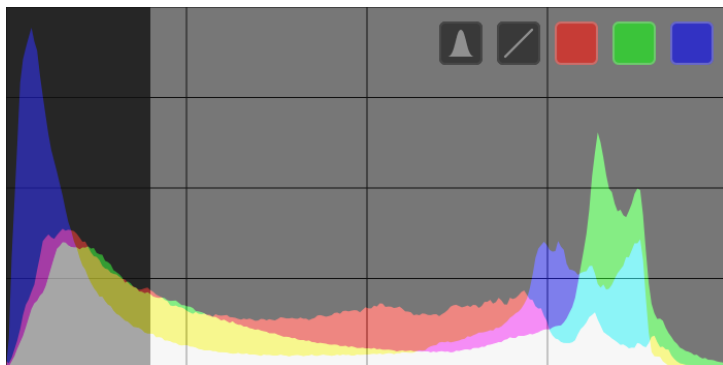
The “preferences...” option in the presets menu allows you to adjust the behavior of the recent collections module as follows:

**number of recent collections to be stored**

The number of recent collections to store and display in the recent collections module (default 10)

### 8.3.5.9. scopes

This module provides various graphical depictions of the developed image’s light levels or [chromaticity](#) .



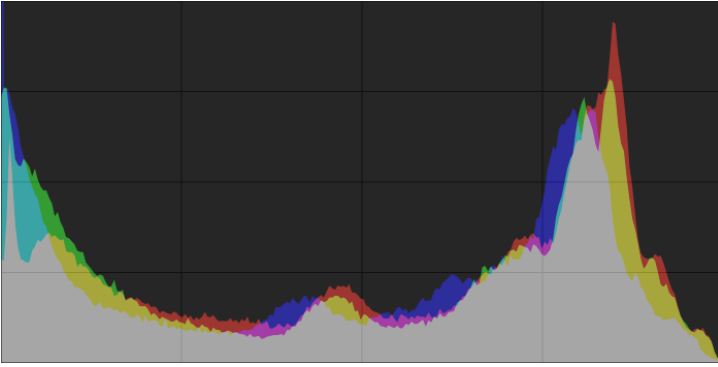
Move the mouse over the panel to show buttons that allow you to adjust the display. The leftmost button cycles the mode between histogram, waveform scope, RGB parade scope, and vectorscope. The remaining buttons control how the plot for the current scope is drawn.

When the mouse is over the scopes panel, scroll with the mouse while holding down the Ctrl key to change the height of the panel. You can show/hide the scopes module entirely with a keyboard shortcut (default Ctrl+Shift+H).

You can move the scopes module to the left-hand panel in [preferences > miscellaneous > position of the scopes module](#) .

For performance reasons, scopes are calculated from the image preview (the image displayed in the [navigation](#) module) rather than the higher quality image displayed in the center view. The preview is calculated at a lower resolution and may use shortcuts to bypass more time-consuming image processing steps. Hence the display may not accurately represent fine detail in the image, and may deviate in other ways from the final developed image.

## histogram



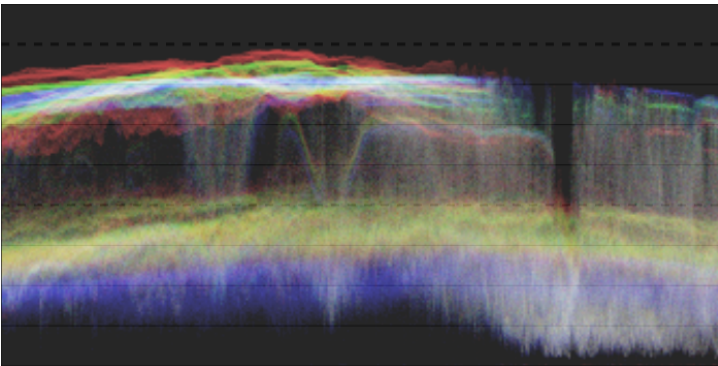
The histogram shows the distribution of pixels by lightness for each color channel.

In its default state, data for all three RGB color channels is displayed. The x-axis runs from 0% to 100% lightness for each channel. The y-axis gives the count of pixels with the given lightness.

Click the second-to-leftmost button on the panel to toggle between a logarithmic and a linear rendering of the y-axis data.

The three rightmost colored buttons toggle the display of the red, green and blue color channels.

## waveform

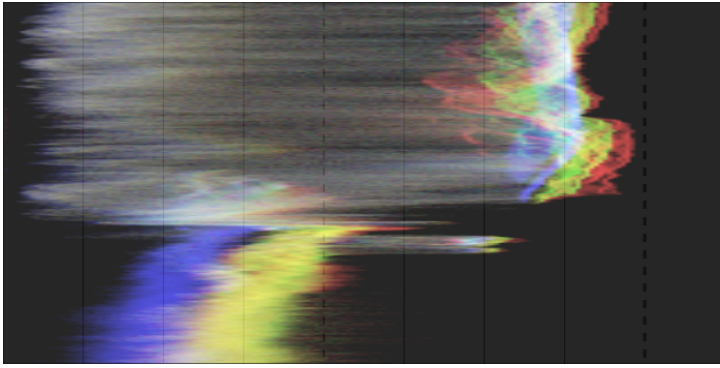


The waveform scope shows similar data to the histogram, but allows you to view that data in a spacial context.

In the “standard” horizontal waveform, the x-axis of the waveform represents the x-axis of the image – the right-hand side of the waveform represents the right-hand side of the image and the left-hand side of the waveform represents the left-hand side of the image.

The y-axis represents the distribution of pixels by lightness for each channel – the dotted line at the top represents 100% lightness (values above this may be clipped), the dotted line in the middle represents 50% lightness, and the bottom of the waveform represents 0% lightness.

The brightness of each point on the waveform represents the number of pixels at the given position (the x-axis) having the given lightness (the y-axis).

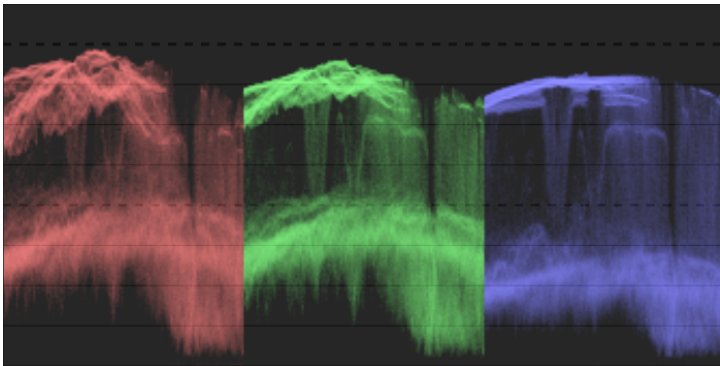


Clicking the second-to-leftmost button on the panel toggles between a horizontal and a vertical waveform. In the vertical waveform, the y-axis of the plot represents the y-axis of the image, and the x-axis represents the distribution of pixels by lightness. The vertical waveform can be useful for portrait-format images, or simply to understand an image in a different way.

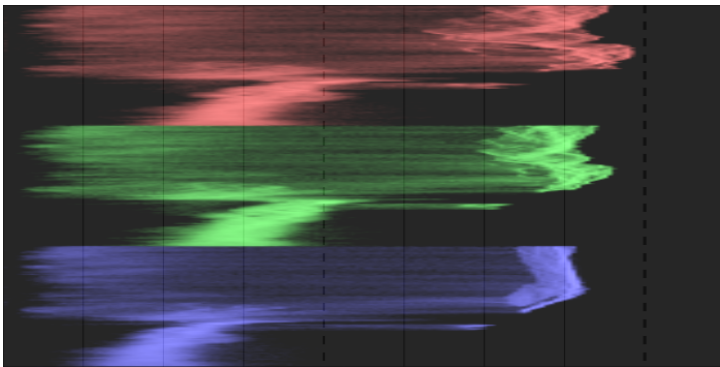
As with the histogram, you can selectively display each of the red, green, and blue channels, by clicking on the appropriate buttons.

See [Of Histograms and Waveforms](#) for more on darktable's waveform scope.

### rgb parade



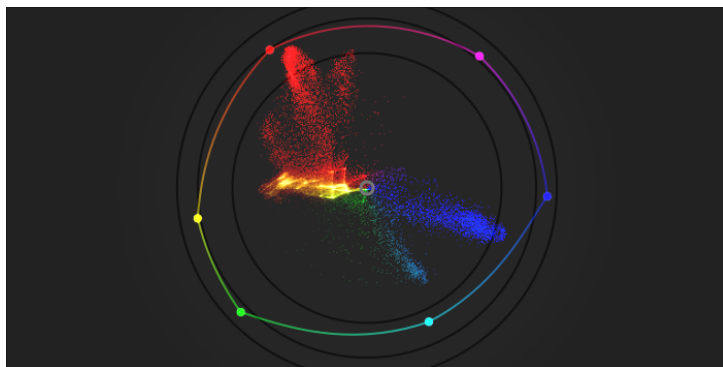
The RGB parade scope shows the same data as the waveform, but with the red, green, and blue channels presented side-by-side.



As with the waveform, clicking the second-to-leftmost button on the panel toggles between horizontal and vertical processing of the image data.

The RGB parade can be useful for matching the intensities of the red, green, and blue channels. It can also help with understanding the differences between and individual qualities of each channel.

### vectorscope



The vectorscope shows [chromaticity](#) without regard to either lightness or spatial data.

The distance from the center of the graph represents chroma and the angle represents hue. Areas of the graph are colored depending on the chromaticity of the color to which they correspond in the image. The graph represents color “volume” by rendering the more frequently used colors in the image in lighter tones.

The vectorscope chroma scale can either be linear or logarithmic. Click the second-to-leftmost button on the panel to toggle this.

The vectorscope can describe the image in the [CIELUV](#) , [JzAzBz](#) or [RYB](#) colorspace. Clicking the third-to-leftmost button on the panel cycles between  $u^*v^*$ , AzBz and RYB. The CIELUV graph will be faster to calculate, and is a well-known standard. JzAzBz may be more perceptually accurate.

The graph includes a “hue ring” representing the maximum chroma of each hue (in bounded RGB) of the current histogram profile. The RGB primaries/secondaries are marked by circles.

### caveats

- The hue ring is not a [gamut check](#) , as a color can be within the hue ring, yet out of gamut due to its darkness/lightness.
- When adjusting an image based upon a color checker, faster and more accurate results will come from using [calibrate with a color checker](#) in the *color calibration* module.
- The vectorscope does not have a “skin tone line”, which is a flawed generalization rather than a universal standard.
- The vectorscope represents a colorimetric encoding of an image, which inevitably diverges from a viewer’s perception of the image.

### exposure adjustment

The histogram, waveform, and RGB parade scopes can be used to directly alter the *exposure* and *black level* of the [exposure](#) module.

For the histogram, click towards the right-hand side of the histogram and then drag right to increase or drag left to decrease the exposure. In a similar manner you can control the black level by clicking and dragging in the left-hand side.

For horizontal waveform and RGB parade, drag the upper portion of the scope up/down to increase/decrease exposure. Drag the lower portion up/down to increase/decrease the black level.

For vertical waveform and RGB parade scopes, the corresponding regions are to the right (exposure) and left (black level).

You can also scroll in the appropriate area – rather than dragging – to adjust exposure and black point. Double-click in the scope to reset the exposure module’s parameters to their defaults.

### histogram profile

Image data is converted to the *histogram profile* before the scope data is calculated. You can choose this profile by right-clicking on the [soft-proof](#) or [gamut check](#) icons in the bottom panel and then selecting the profile of interest. When soft-proof or gamut check is enabled, the scope is shown in the soft proof profile.

As the scopes module runs at the end of the preview pipeline, it receives data in display color space. If you are using a display color space that is not “well behaved” (this is common for a device profile), then colors that are outside of the gamut of the display profile may be clipped or distorted.

### 8.3.5.10. tagging

Manage tags attached to images.

Tags provide a means of adding information to images using a keyword dictionary. You can also manage tags as a hierarchical tree, which can be useful when their number becomes large.

Tags are physically stored in [XMP sidecar files](#) as well as in darktable's library database and can be included in [exported](#) images.

#### definitions

The following definitions assume that you have set up a single tag named "places|France|Nord|Lille".

#### **tag**

A tag is a descriptive string that may be attached to an image. A tag can be a single term or a sequence of connected terms forming a path, separated by the pipe symbol. For example, "places|France|Nord|Lille" defines a single tag, where each term in the path forms a smaller subset of those before it. You can attach as many tags to an image as you like.

You can assign properties (name, private, category, synonyms and image order) to a tag.

#### **node**

Any path that forms part of a tag is a node. In the above example, "places", "places|France", "places|France|Nord" and "places|France|Nord|Lille" are all nodes. In the hierarchical tree view, the nodes form the branches and leaves of the tree.

#### **free node**

Any node that is not explicitly defined as a tag is called a free node. In the above example, "places", "places|France" and "places|France|Nord" are all free nodes. You cannot set any properties, except "name", for a free node and you cannot add a free node to an image. See the "multiple tags" section below for more information.

#### **category**

Any tag can be flagged by the user as being a "category".

#### multiple tags

The above definitions considered a simple example - a single tag and its properties. Consider instead the scenario where the following four pipe-delimited tags are each separately defined in darktable.

```
places|France|Nord|Lille
places|France|Nord
places|France
places|England|London
```

In this case the nodes are

```
places
places|France
places|France|Nord
places|France|Nord|Lille
places|England
places|England|London
```

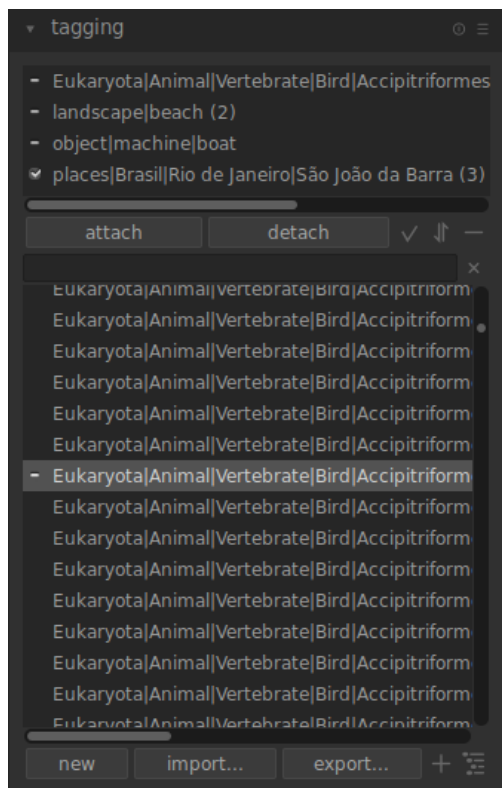
The only free nodes are "places" and "places|England". Both of these free nodes are also (by implication) categories.

You can attach any of these tags to any image. Any tags attached to an image, except category tags, can be included when that image is [exported](#).

If you attach the "places|France|Nord|Lille" tag to an image, the "places|France|Nord" and "places|France" tags are also implicitly attached to that image (you don't need to attach them manually). Note that this is only true here because those additional tags have been separately defined - the "places" node is not included because it is a "free node" (not a tag).

#### module sections





The tagging module consists of two sections

1. The upper *attached tags* section (with the *attach/detach* buttons under it)
2. The lower *tag dictionary* section (with the *new/import.../export...* buttons under it)

### attached tags section

The *attached tags* section displays tag(s) attached to image(s), where those images are

- under your mouse cursor (if hovering over an image in the lighttable view); or
- currently selected (if not hovering over an image)

At the bottom of the *attached tags* section are the following buttons, from left to right:

#### **attach**

If a tag is selected in the *tag dictionary* section, attach this tag to the selected images.

#### **detach**

If a tag is selected in the *attached tags* list, detach that tag from the selected images. A tag can also be detached if you right click on the tag name and select *detach* from the pop-up menu.

#### **hidden tags**

Choose whether to view any hidden tags that darktable has automatically attached to the selected images.

#### **sort**

Choose whether to sort the *attached tags* list alphabetically or by the count shown in brackets next to the tag (this count indicates how many of the selected images have that tag attached to them).

#### **parents**

Choose whether or not to show the parent categories of the tag.

You can adjust the height of the *attached tags* window by holding Ctrl and scrolling with your mouse wheel.

## tag dictionary section

The *tag dictionary* section displays all of the tags that are available in darktable's database. At the top of the *tag dictionary* section is a text box where tag names can be entered. Below this is a list of available tags, which may also include indicator symbols to the left of the tag names. The meanings of these symbols are as follows:

- a check mark [✓] indicates that the tag is attached to all of the selected images
- a minus sign [-] indicates that the tag is attached to at least one of the selected images. If the symbol is next to a node name in the hierarchical *tree* view, it means that one of the child tags under that node is attached to at least one of the selected images.
- if no indicator symbol is present, this means that the tag is not attached to any of the selected images, or that the node has no child tags attached to any of the selected images.

In the hierarchical *tree* view, a name in italics represents either a free node or a category.

Below the *tag dictionary* list are the following buttons, from left to right:

### **new**

Create a new tag, using the name that has been entered into the text entry box at the top of the *tag dictionary* section.

### **import...**

Import tags from a Lightroom keyword file.

### **export...**

Export all tags to a Lightroom keyword file.



### **suggestions**

Show a list of suggested keywords based on the keywords already associated with the selected images. CAUTION: This view queries the database so might be slow.



### **list/tree**

Toggle the display of tags between the straight *list* view and hierarchical *tree* view.

You can adjust the height of the *tag dictionary* window by holding Ctrl while scrolling with your mouse wheel.

## usage

The following sections describe the operations that can be performed with tags.

### text entry

The text entry box (shown under the *attach/detach* buttons) has multiple purposes.

- If the *tag dictionary* list is in *list* view mode (rather than *tree* view mode), then typing the first few characters of a tag will bring up a list of suggestions. You can then scroll down with the arrow keys and press Enter to select one of the suggestions. Pressing Enter a second time will attach it to the selected images. You can also edit the name of the tag before pressing Enter – in this case the tag will be created if it doesn't already exist in the database.
- Aside from the auto-completion suggestions, typing some partial text into the text entry box allows you filters the set of tags shown in the *tag dictionary* window to those whose name or synonym matches the entered text. Press Enter to attach a tag with the entered name to the selected images. If that tag name does not yet exist in the database, it will be created before being attached.
- The pop-up menu entry “copy to entry” can be used to copy a selected tag to the text entry box. You can then edit this name and press Enter to create a new tag with that name, making it easy to create multiple tags with similar names.

---

**Note:** If you find the auto-completion behaviour annoying, it can be disabled in [preferences > miscellaneous > tags > disable the entry completion](#) .

---

## create tag

There are several ways to create a new tag:

- *Import a text file.* You can import one or more text files in the Lightroom tagging file format. You can also export your tags, edit the exported file, then re-import it. The import function updates existing tags and creates new tags as required. If you change the name of a tag in an imported file, it will be treated as a new tag.
- *Import already-tagged images.* This method does not offer any flexibility to change tag names or categories during the import process.
- *Use the “create tag” sub-menu.* A tag can be created manually, under an existing one (hierarchical) or at the root level.
- *Use the “set as a tag” sub-menu.* You can set a free node (e.g. “places|England”) as a tag so that it gets implicitly attached to all images tagged with its sub-tags (e.g. “places|England|London”).
- *Type into the text box and press the “new” button.* Hierarchical tags are created using the pipe symbol “|” to separate nodes. Note that the entered tag is also attached to any selected images.

A number of tags are automatically generated by darktable when certain actions are undertaken. For example, the tags “darktable|exported” and “darktable|styles|your\_style” can be used to identify images that have been exported and had styles applied, respectively.

## edit/rename tag

The *tag dictionary* can be maintained via the “edit...” and “change path...” items in the right-click pop-up menu.

The “edit...” operation allows you to change the name of a tag, though you cannot change which node it belongs to (you cannot use the pipe “|” symbol in the tag name field). The command is aborted if you try to enter a tag name that already exists. You can set the *private* and *category* flags and define *synonyms* for the tag (see below). These attributes are recorded in the XMP-dc Subject and XMP-lr Hierarchical Subject metadata entries, respectively. You can control which tags are included in exports by changing settings in the [export](#) module.

- A tag set as “private” is, by default, not exported.
- A tag set as “category” is not exported in XMP-dc Subject. However it is exported in XMP-lr Hierarchical Subject as this XMP metadata holds the organization of your tags.
- “synonyms” enrich the tag information and are mainly used to assist search engines. For example “juvenile”, “kid” or “youth” can be set as synonyms of “child”. Synonyms can also be used to translate tag names to other languages.

The “change path...” operation is only available in the *tree* view mode, and it shows the number of tagged images which would be impacted by a change to the name of this node. The change path window lets the user change the full path of the node, including the nodes to which it belongs (nodes can be specified using the pipe “|” symbol). This operation is powerful, but please take care as it can have a significant impact on the metadata of your images. The operation is aborted if the requested change causes a conflict with an existing tag.

A quick way to organize the tag structure is to drag and drop the nodes. In the *tree* view mode, you can drag any node and drop it on top of any other node. The first node and its descendants, if any, become descendants of the second node. Dragging over a node automatically opens that node. To place a node at the root level, drag it onto the top of the tagging window. If the requested change causes a conflict with an existing tag, the operation is aborted.

## attach tag

There are a number of ways to attach an existing tag to a group of selected images:

- click on a tag in the *tag dictionary* window to select it, then click on the *attach* button.
- right-click on a tag in the *tag dictionary* window, to show a pop-up menu, then select the “attach tag” menu item.
- double-click on a tag in the *tag dictionary* window.
- right-click on a tag shown in the *attached tags* view to show a pop-up menu. If some of the selected images do not currently have that tag, the “attach tag to all” menu item can be used to attach that tag to all the selected images.
- Type into the text box and press the “new” button or the Enter key. This will create the tag if it doesn’t already exist, and attach it to the selected images.
- Press Ctrl+T to open a small text box at the bottom of the central view of the lighttable. Type in the name of a tag and press Enter. The tag will be created if it doesn’t exist, and attached to all the selected images.
- Drag an image or group of images and drop it onto the desired tag.

When hovering over the images in the lighttable you can check which tags are attached to the image, either by looking at the *attached tags* window in the *tagging* module, or in the *tags* attribute in the [image information](#) module.

## detach tag

There are several ways to remove a tag from a group of selected images:

- click on a tag in the *attached tag* window of the *tagging* module to select the tag, then click on the *detach* button.
- double-click on a tag in the *attached tag* window.
- right-click on a tag in the *attached tag* window, to show a pop-up menu, and select “detach”.

## delete tag

It is possible to completely remove a tag from all images (whether selected or not) and delete the tag from the database. Because this could impact a large number of images, a warning will be displayed indicating how many images currently have this tag attached. Take this warning seriously as there is no way to undo this action (except by restoring your database and/or XMP sidecar files from a backup). A tag in *tag dictionary* window can be deleted in the following ways:

- right-click on a tag in the *tag dictionary* window, to show a pop-up menu, and choose “delete tag”.
- right-click on a branch node in the *tag dictionary* window, to show a pop-up menu, and choose “delete node” to delete the selected node together with any child nodes.

## import / export

The “import” button allows you to choose a text file (which must comply with the Lightroom tag text file format) and import its content. If a tag in the imported file already exists, its properties will be updated, otherwise a new tag will be created.

The “export” button exports your entire tag dictionary into a text file of your choice (Lightroom tag text file format).

## navigation

To see the images bearing a particular tag in the *tag dictionary* window, right-click on the tag name and choose “go to tag collection” in the resulting pop-up menu. This opens a collection in the [collections](#) module showing all images containing this tag. You can also select other tags in the *collections* module by double-clicking on the collection for the other tag.

To return to the collection that was selected before opening a tag collection select the “go back to work” item from the pop-up menu. This will allow you to return to the original collection, as long as you haven’t selected any other collections in the meantime.

## 8.3.6. tethering

### 8.3.6.1. camera settings

Set up an image capture job.

This can include sequence, bracket and delayed captures. You are also able to control other camera settings such as focus mode, aperture, shutter speed, ISO and white balance.

### 8.3.6.2. live view

Control your camera's live view mode.

Functionality such as focus control, rotation, guides and overlays are supported.

### 8.3.6.3. session

Set a session jobcode to allow the creation of a new film roll.

A session is a sequence of exposures taken in tethering mode and placed into a single film roll. A new session means the creation of a new film roll.

The session module allows you to set a jobcode, which can be used to create your new film roll. See [preferences > import > session options](#) for details about how to create new film rolls using the \$(JOBCODE) variable in the session module.

# 9. Preferences & Settings

## 9.1. overview

darktable comes with a number of user-configurable preferences. These can be adjusted in the preferences dialog, which can be reached by clicking the gear icon on the right of the [top panel](#) . Preferences are separated into tabs, each of which is described in more detail in the following sections.

When opened from within the lighttable or darkroom views, the appropriate tab will be loaded by default to allow you to modify that view's settings.

Any altered settings (i.e. those that differ from their default state) are highlighted with a bullet beside them. If you change a setting that needs a restart to take effect, a message will appear as a reminder after you exit the preferences dialog.

Double click on a setting's label to reset to its default value.

The preferences dialog can be closed by clicking on the close button in your window manager or pressing the Escape key.

## 9.2. general

Control the overall look and feel of darktable.

### **interface language**

Set the language of the user interface. The system default is marked with an \* (needs a restart)

### **theme**

Set the theme for the user interface. Aside from any aesthetic considerations, the recommended interface color for color evaluation is middle gray. Visual perception is affected by ambient brightness, and a low user interface brightness causes all kinds of illusions. Using a dark interface to retouch photos can therefore lead to excessive retouching (abuse of contrast and saturation) and to a photo that is too dark when printed. It is therefore highly recommended that you use one of the "gray" themes for retouching work as these are designed so that the user interface approximates middle gray (default "darktable").

### **prefer performance over quality**

Enable this option to render thumbnails and previews at a lower quality. This increases the rendering speed by a factor of 4, and is useful when working on slower computers (default off). This also improves the performance of slideshow image rendering.

### **use system font size**

Select this option to use the font size defined by your system. If unchecked, you may enter a custom font size in the box below (default on).

### **font size in points**

If the "use system font size" option is switched off, enter a font size (in points) for darktable to use. The font size will be changed immediately.

### **GUI controls and text DPI**

Adjust the global GUI resolution to rescale controls, buttons, labels, etc. Increase for a magnified GUI, decrease to fit more content in the window. Set to -1 to use the system-defined global resolution. The default is 96 DPI on most systems. (needs a restart)

## CSS Theme Modifications

In addition to selecting a pre-built theme you can also apply additional CSS customisations of your own to tweak the look-and-feel of darktable. A text box is provided for this purpose.

When you have finished entering your CSS tweaks, click on the 'save CSS and apply' button. This will save your CSS to a file (\$HOME/.config/darktable/user.css) and immediately apply it to the current darktable session.

If your changes cause any issues, you can uncheck the "modify selected theme with CSS tweaks below" check box. This will immediately restore the base theme but will leave your tweaks in the editor so that you can re-edit them and try again. Simply press "save CSS and apply" again when you are ready to retry. This will automatically re-check the "modify selected theme with CSS tweaks below" checkbox and apply the new CSS.

If you have any issues with darktable please retry with this option unchecked to be certain that they were not caused by any of your alterations.

If you want to understand how darktable themes are constructed you are advised to look at the file \$DARKTABLE/share/darktable/themes/darktable.css for more information (where \$DARKTABLE is darktable's installation directory). You can also use the Gtk Inspector tool to analyse the existing layout.

## 9.3. import

Control default file naming conventions used when [importing](#) images.

### session options

The following options define the default naming pattern for use in the "copy & import" or "copy & import from camera" options in the import module, or when taking photos in the [tethering](#) view.

The naming pattern consists of three parts: a base part defining the parent folder, a session part defining a sub directory (which is specific to the individual import session), and a file name part defining the filename structure for each imported image.

Several pre-defined variables can be used in the pattern as placeholders:

\$(HOME)	the home folder as defined by the system
\$(PICTURES_FOLDER)	the pictures folder as defined by the system (usually "\$HOME/Pictures")
\$(DESKTOP)	the desktop folder as defined by the system (usually "\$HOME/Desktop")
\$(USERNAME)	your user account name on the system
\$(FILE_NAME)	basename of the imported image
\$(FILE_EXTENSION)	extension of the imported image
\$(JOBCODE)	unique identifier of the import job
\$(SEQUENCE)	a sequence number within the import job
\$(MAX_WIDTH)	maximum image width to limit within export session
\$(MAX_HEIGHT)	maximum image height to limit within export session
\$(ID)	unique identification number of the image in darktable's database
\$(YEAR)	year at the date of import
\$(MONTH)	month at the date of import
\$(DAY)	day at the date of import
\$(HOUR)	hour at the time of import
\$(MINUTE)	minute at the time of import
\$(SECOND)	second at the time of import
\$(EXIF_YEAR)	year the photo was taken (from Exif data)
\$(EXIF_MONTH)	month the photo was taken (from Exif data)
\$(EXIF_DAY)	day the photo was taken (from Exif data)
\$(EXIF_HOUR)	hour the photo was taken (from Exif data)
\$(EXIF_MINUTE)	minute the photo was taken (from Exif data)
\$(EXIF_SECOND)	seconds the photo was taken (from Exif data)
\$(EXIF_ISO)	ISO value of the photo (from Exif data)



**base directory naming pattern**

The base directory part of the naming pattern (default \$(PICTURES\_FOLDER)/Darktable).

**sub directory naming pattern**

The sub directory part of the naming pattern (default \$(YEAR)\$ (MONTH)\$ (DAY)\_\$(JOBCODE)).

**keep original filename**

Check this box to keep original the filename instead of using the pattern below when importing from a camera or card (default off).

**file naming pattern**

The file name part of the naming pattern (default \$(YEAR)\$ (MONTH)\$ (DAY)\_\$(SEQUENCE) . \$(FILE\_EXTENSION)).

## 9.4. lighttable

Control functionality in the [lighttable](#) view and modules.

### general

**hide built-in presets for utility modules**

If enabled, only user-defined presets will be shown in presets menu for utility modules – built-in presets will be hidden (default off).

**use single-click in the collections module**

Enable “single click” mode in the [collections](#) module, which allows ranges to be selected (default off).

**expand a single utility module at a time**

Controls how utility modules are expanded. If this option is enabled, expanding a module by clicking collapses any other currently expanded panel. If you want to expand a panel without collapsing the others you can do so with Shift+click. Disabling this option inverts the meaning of click and Shift+click (default off).

**scroll to utility modules when expanded/collapsed**

With this option enabled the side panels will scroll a utility module to the top of the panel when it is expanded or collapsed (default off).

**rating an image one star twice will not zero out the rating**

Normally clicking a one star rating twice will set a zero star rating to that image. Check this option to disable this functionality (default off).

**show scrollbars for center view**

Choose whether to show scrollbars in the center view of the lighttable (default on).

### thumbnails

**use raw file instead of embedded JPEG from size**

When generating thumbnails for images that have not yet been processed in the darkroom, if the thumbnail size is greater than this value, generate it by processing the raw image data. If the thumbnail is below this size, use the JPEG preview image embedded in the raw file. Once an image has been processed in the darkroom, thumbnails will always be generated from raw data (you can revert back to the JPEG preview by discarding history). To render thumbnails with the best quality choose “always”.

**high quality processing from size**

If the thumbnail size is greater than this value and is being generated from raw data, it will be processed using the full quality rendering path, which is better but slower (default 720p). To render thumbnails with the best quality, choose “always”.

**delimiters for size categories**

Size categories are used to allow different thumbnail overlays to be shown depending on the thumbnail size. A pipe delimited set of values defines at what image sizes the category changes. The default value of “120|400” means that there will be 3 categories of thumbnail: 0-120px, 120-400px and >400px.

**pattern for the thumbnail extended overlay text**

If you have chosen to show extended overlay text over thumbnail images, this setting allows you to define what information is displayed. This pattern can use any of the variables defined in the [variables](#) section.

**pattern for the thumbnail tooltip (empty to disable)**

Defines what information is displayed in the tooltip when the mouse hovers over image thumbnails. This pattern can also use any of the variables defined in the [variables](#) section.

## 9.5. darkroom

Control functionality in the [darkroom](#) view and associated modules.

### general

**pen pressure control for brush masks**

Controls how the pressure reading of a graphics tablet impacts newly generated [drawn mask](#) brush strokes. You can control the brush width, hardness and opacity. “Absolute” control means that the pressure reading directly defines the attribute with a value between 0% and 100%. “Relative” means that the pressure reading adjusts the attribute between zero and the pre-defined default value (default off).

**smoothing of brush strokes**

Sets the level for smoothing of [drawn mask](#) brush strokes. Stronger smoothing leads to fewer nodes and easier editing at the expense of lower accuracy.

**scroll down to increase mask parameters**

By default, scrolling your mouse up increases the value of the relevant shape parameters in [drawn masks](#) . Set this preference to reverse the behavior (default off).

**pattern for the image information line**

Set the information to be displayed in the [image information line](#) . You can use any variables in the [variables](#) section as well as \$(NL) for a new line.

**position of the image information line**

Choose the darkroom panel in which the [image information line](#) is displayed. Choose between “top left” “top right” “top center” “bottom” and “hidden” (default “bottom”).

**border around image in darkroom mode**

Display the center image in darkroom mode with an outside border of the given number of pixels (default 20).

**show scrollbars for center view**

Choose whether to show scrollbars in the center view of the darkroom (default off).

**demosaiing for zoomed out darkroom mode**

Choose how to demosaic images in the darkroom view when not viewing the image at 1:1 zoom scale. Note: this setting is only used for interactive work and does not affect exported images.

- *always bilinear (fast)* is fastest, but not as sharp
- *at most RCD (reasonable)* uses RCD + interpolation modes. X-Trans sensors will use VNG as the middle-ground setting.
- *full (possibly slow)* will use the settings from the [demosaic](#) module.

(default “*at most RCD (reasonable)*”).

**reduce resolution of preview image**

Reduce the resolution of the [navigation preview](#) image (choose from “original”, “1/2”, “1/3” or “1/4” size). This may improve the speed of the rendering but take care as it can also hinder accurate color-picking and masking (default “original”).

**show loading screen between images**

Show gray loading screen when navigating between images in the darkroom. Switch this option off to just show a simple toast message and leave the previous image in place until the next image is loaded. Note that switching this option off can be very useful to quickly compare duplicate images, however, there might be issues with long loading times (leading you to think the next image has already loaded) and you may observe visual artifacts while the next image is loading (default on).

### modules

**display of individual color channels**

Control how individual color channels are displayed when activated in the [parametric masks](#) feature. You can choose between “false color” and “gray scale” (default “false color”).

**hide built-in presets for processing modules**

If enabled, only user-defined presets will be shown in the presets menu for processing modules – built-in presets will be hidden (default off).

**expand a single processing module at a time**

Control how [processing modules](#) are expanded in the darkroom. If this option is enabled, expanding a module by clicking collapses any currently expanded module. If you want to expand a module without collapsing the others you can do so with Shift+click. Disabling this option inverts the meaning of click and Shift+click (default on).

**only collapse modules in current group**

When choosing to expand a single processing module at a time (using the logic defined in the previous setting), only collapse other modules that appear in the current visible group. Disable this option to ensure that modules in non-visible groups are also collapsed (default on).

**expand the module when it is activated, and collapse it when disabled**

Select this option for the darkroom to automatically expand or collapse [processing modules](#) when they are enabled or disabled. (default off)

**scroll to processing modules when expanded/collapsed**

With this option enabled [processing modules](#) are scrolled to the top of the right-hand panel when expanded or collapsed (default on).

**white balance slider colors**

Controls the appearance of the temperature/tint sliders in the [white balance](#) module.

- *no color*: background of the sliders is not colored at all.
- *illuminant color*: slider colors represent the color of the light source, i.e. the color you are adjusting to in order to achieve neutral white
- *effect emulation*: slider colors represent the effect the adjustment would have had on the scene. This is how most other raw processors show temperature/tint sliders colors.

(default *no color*)

**colorbalance slider block layout**

Controls the appearance of the shadows/mid-tones/highlights sections in the [color balance](#) module.

- *list*: all sliders are shown in one long list (with headers)
- *tabs*: use tabs to switch between the blocks of sliders
- *columns*: the blocks of sliders are shown next to each other (in narrow columns)

(default *list*)

**show right-side buttons in processing module headers**

Choose whether to show the four buttons (mask indicator, multi-instance menu, reset, presets menu) on the right-hand-side of the [module header](#) for processing modules. These buttons will always appear when the mouse is over the module. At other times they will be shown or hidden according to this preference selection:

- *always*: always show all buttons
- *active*: only show the buttons when the mouse is over the module
- *dim*: buttons are dimmed when mouse is not over the module
- *auto*: hide the buttons when the panel is narrow
- *fade*: fade out all buttons when the panel narrows
- *fit*: hide all the buttons if the module name doesn't fit
- *smooth*: fade out all buttons in one header simultaneously
- *glide*: gradually hide individual buttons as needed

(default *always*)

**show mask indicator in module headers**

If enabled, an icon will be shown in the header of any processing modules that have a [mask](#) applied (default on).

**prompt for name on addition of new instance**

If enabled, when creating a new instance of a processing module, a prompt will be immediately displayed allowing you to set a name for the new instance (default off).

## 9.6. processing

Control how images are processed.

# image processing

## always use LittleCMS 2 to apply output color profile

If this option is activated, darktable will use the LittleCMS 2 system library to apply the output color profile instead of its own internal routines. This is significantly slower than the default but might give more accurate results in some cases.

If the given ICC is LUT-based or contains both a LUT and a matrix, darktable will use LittleCMS 2 to render the colors regardless of this parameter's value (default off).

## pixel interpolator (warp)

The pixel interpolator used for rotation, lens correction, liquify, crop and final scaling.

Whenever we scale or distort an image we have to choose a pixel interpolation algorithm (see [wikipedia](#) for details). For warping modules, darktable offers bilinear, bicubic or lanczos2. In general, bicubic is a safe option for most cases and is the default value.

## pixel interpolator (scaling)

The pixel interpolator used for scaling. The same options are provided as for the warp modules, but with the addition of lanczos3.

lanczos3 can cause pixel overshoots leading to artefacts but sometimes gives a more crisp visual appearance. This option is therefore only provided for transforming (scaling) algorithms and is the default value.

## 3D lut root folder

Define the root folder (and sub-folders) containing Lut files used by the [lut 3D](#) module

## auto-apply pixel workflow defaults

Choose which modules and module order to apply to new images by default:

- *scene-referred* workflow (default) assumes that most processing will be performed in a linear RGB color space. Selecting this option will automatically enable the [filmic rgb](#) and [exposure](#) modules and will set the pixelpipe order to *v3.0* which is the (scene-referred) order defined for darktable 3.0 and later.

The *exposure* module will include an automatic adjustment of +0.5 EV to adjust the mid-gray to match that of the majority of SLR cameras. This adjustment can be overridden with an automatically-applied preset if the default produces consistently dark images for your camera.

Finally, this setting automatically enables the “compensate camera exposure” option in the *exposure* module to adjust the global brightness appropriately in cases where the camera's exposure compensation dial was used.

- *display-referred* workflow assumes that most processing will be performed in the Lab color space and is the legacy mode from darktable 2.6 and earlier. Selecting this option will automatically enable the [base curve](#) module and set the pixelpipe order to the *legacy* (display-referred) order used by default up to version 2.6.
- *none* will not enable any modules by default and will set the pixelpipe to the *v3.0* (scene-referred) order defined for darktable 3.0 and later.

## auto-apply chromatic adaptation defaults

Choose which module is responsible for performing white balance adjustments (chromatic adaptation) by default. Select “legacy” (default) to perform basic chromatic adaptation within the [white balance](#) module only. Select “modern” to use a combination of the *white balance* and [color calibration](#) modules to perform modern chromatic adaptation with improved color science. These settings are applied by default to new edits and will not impact old edits.

## auto-apply per camera basecurve presets

Use a per-camera base curve by default (if available) instead of the generic manufacturer one. This should only be used in conjunction with the *display-referred* workflow defined above (default off).

## auto-apply sharpen

Auto-apply the sharpen module to new images by default. This option is not recommended on cameras without a low-pass filter. (default on, requires a restart).

## detect monochrome previews

Enable this option to analyse images during import and tag them with the `darkroom|mode|monochrome` tag if they are found to be monochrome. The analysis is based on the preview image embedded within the imported file. This makes for a more convenient workflow when working with monochrome images, but it slows down the import, so this setting is disabled by default.

**show warning messages**

Enable this option to display warning messages in processing modules where non-standard and possibly harmful settings have been used in the pipeline. Such messages can sometimes be false-positives (because of intentional non-standard settings) and can be disregarded if you know what you are doing. Disable to hide these warnings. (default on).

## cpu / gpu / memory

**memory in megabytes to use for thumbnail cache**

In order to speed up the display of film rolls, darktable stores image thumbnails in a cache on disk (primary cache) and loads it into memory at startup. This setting controls the size of the cache in megabytes. Needs a restart if changed (default 512MB).

**enable disk backend for thumbnail cache**

If activated, darktable stores all thumbnails on disk as a secondary cache, and thereby keeps thumbnails accessible if they are dropped from the primary cache. This needs more disk space but speeds up the [lighttable](#) view as it avoids the reprocessing of thumbnails (default on).

**enable disk backend for full preview cache**

If enabled, darktable writes full preview images to disk (`.cache/darktable/`) when evicted from the memory cache. Note that this can take a lot of storage (several gigabytes for 20k images) and darktable will never delete cached images. It's safe to delete these manually if you want. Enabling this option will greatly improve lighttable performance when zooming an image in full preview mode (default off).

**host memory limit (in MB) for tiling**

In order to manage large images on systems with limited memory darktable does tile-wise processing. This variable controls the maximum amount of memory (in MB) a module may use during image processing. Lower values will force memory-hungry modules to process an image with increasing number of tiles. Setting this to 0 will omit any limits. Values below 500 will be treated as 500. Needs a restart if changed (default 1500).

**minimum amount of memory (in MB) for a single buffer in tiling**

If set to a positive, non-zero value, this variable defines the minimum amount of memory (in MB) that darktable should take for a single tile. Needs a restart if changed (default 16).

**activate [OpenCL](#) support**

*darktable* can use your GPU to significantly speed up processing. The OpenCL interface requires suitable hardware and matching OpenCL drivers on your system. If one of those is not found the option is grayed out. Can be switched on and off at any time and takes immediate effect (default on).

**OpenCL scheduling profile**

Defines how preview and full pixelpipe tasks are scheduled on OpenCL enabled systems:

- *default*: the GPU processes the center view pixelpipe; the CPU processes the preview pipe;
- *multiple GPUs*: both pixelpipes are processed in parallel on two different GPUs;
- *very fast GPU*: both pixelpipes are processed sequentially on the GPU.

## 9.7. security

Control whether warning messages are shown before undertaking certain activities.

### general

**ask before removing images from the library**

Always ask before removing an image from darktable's library database, where the image/xmp file is retained (default on).

**ask before deleting images from disk**

Always ask before deleting an image file (default on).

**ask before discarding history stack**

Always ask before discarding the history stack of an image (default on).

**try to use trash when deleting images**

Instead of physically deleting images from disk, attempt to put them into the system's trash bin (default on).

**ask before moving images from film roll folder**

Always ask before moving an image file (default on).

**ask before copying images to new film roll folder**

Always ask before copying an image file to a new location (default on).

**ask before removing empty folders**

Always ask before removing any empty folder. This can happen after moving or deleting images (default off).

**ask before deleting a tag**

Always ask before deleting a tag from an image (default on).

**ask before deleting a style**

Always ask before deleting a style (default on).

**ask before deleting a preset**

Always ask before deleting a preset (default on).

**ask before exporting in overwrite mode**

Always ask before [exporting](#) images in overwrite mode.

## other

**password storage backend to use**

The backend to use for password storage. Options: “auto” (default), “none”, “libsecret”, “kwallet”.

**executable for playing audio files**

Define an external program which is used in the lighttable view to play audio files that some cameras record to keep notes for images (default “aplay”).

## 9.8. storage

The following options are related to darktable’s library database and [XMP sidecar files](#) .

## database

**check for database maintenance**

Indicates when darktable should check for database fragmentation and perform maintenance. Options are “never”, “on startup”, “on close” and “on both”. Each of these is also available with an additional “(don’t ask)” option to perform the checks automatically without prompting (default “on close”).

**database fragmentation ratio threshold**

Fragmentation ratio (in per cent) above which database maintenance should be performed (subject to the selection made in the option above) (default 25).

**create database snapshot**

Specifies how often darktable should create database snapshots. Options are “never”, “once a month”, “once a week”, “once a day” and “on close” (default “once a week”)

**how many snapshots to keep**

Number of snapshots to keep after creating a new snapshot, not counting database backups taken when moving between darktable versions. Enter “-1” to store an unlimited number of snapshots. (default 10)

## xmp

### write sidecar file for each image

XMP files provide a redundant method of saving the changes that you have made to an image, in addition to the changes saved to darktable's database. This option allows you to choose when to write these files. Choose from:

- *never*: Don't write XMP files. This can be useful if you are running multiple version of darktable for development/testing purposes but is not normally recommended
- *on import*: An XMP file will be written as soon as you add an image to darktable's library and will be subsequently updated each time you edit it
- *after edit*: An XMP file will be written the first time you perform an edit on an image and will be updated on each subsequent edit. Importing will not generate an XMP file by default.

It's strongly recommended that you choose either "on import" or "after edit". XMP files provide a useful fail-safe so that you don't lose data if your database becomes corrupted. Backing up your raw file plus the accompanying XMP file will allow you to fully restore your work at a later date by re-importing your edit history back into darktable (default "on import").

### store xmp tags in compressed format

Entries in XMP tags can get rather large and may exceed the available space to store the history stack in some output files on export. This option allows binary XMP tags to be compressed in order to save space. Available options are "never", "always", and "only large entries" (default).

### look for updated xmp files on startup

Scan all XMP files on startup and check if any have been updated in the meantime by some other software. If updated XMP files are found, a menu is opened for the user to choose which of the XMP files to reload (replacing darktable's database entries by the XMP file contents) and which of the XMP to overwrite from darktable's database. Activating this option also causes darktable to check for text sidecar files that have been added after import time (default off).

## 9.9. miscellaneous

## interface

### sort built-in presets first

Choose how the presets menu is sorted. If this option is enabled, built-in presets are shown first. If the option is disabled, user presets are shown first (default on).

### mouse wheel scrolls modules side panel by default

When enabled, the mouse wheel scrolls side panels by default and Ctrl+Alt+wheel scrolls data entry fields. When disabled, this behavior is reversed (default off).

### always show panels' scrollbars

Defines whether the panel scrollbars should be always visible or activated only depending on the panel content (default on). (needs a restart)

### position of the scopes module

Choose whether to show the [scopes](#) module in the left or right panel (default right). (needs a restart)

### method to use for getting the display profile

This option allows the user to force darktable to use a specific method to obtain the current display profile for [color management](#). In the default setting "all", darktable will choose to query the X display server's xatom or the colord system service. You can set this option to "xatom" or "colord" to enforce a specific method if the two methods give different results. You can run the [darktable-cmstest](#) binary to examine your color management subsystem.

## tags

### omit hierarchy in simple tag lists

When exporting images any hierarchical tags are also added as a simple list of non-hierarchical tags to make them visible to some other programs. When this option is checked darktable will only include the last part of the hierarchy and ignore the rest. So foo|bar|baz will only add baz.



**disable the entry completion**

The entry completion is useful for those who enter tags using the keyboard only. For others the entry completion can be embarrassing (default off). (needs a restart)

## keyboard shortcuts with multiple instances

It is possible to create multiple instances of many processing modules. In this scenario it is not always obvious which instance should be controlled by keyboard shortcut operations. The following options control rules that are applied (in order) to decide which module instance keyboard shortcuts should be applied to.

These rules are also used when clicking and dragging on the scopes module to change exposure.

**prefer expanded instances**

If instances of the module are expanded, ignore collapsed instances (default off).

**prefer enabled instances**

After applying the above rule, if remaining instances of the module are active, ignore inactive instances (default off).

**prefer unmasked instances**

After applying the above rules, if remaining instances of the module are unmasked, ignore masked instances (default off).

**selection order**

After applying the above rules, apply the shortcut to the first or last instance remaining (default "last instance").

## map / geolocalisation view

**pretty print the image location**

Show a more readable representation of the geo-location in the [image information](#) module (default on).

## slideshow view

**waiting time before each picture in slideshow**

The number of seconds before displaying the next image (default 5) in the [slideshow view](#) .

## 9.10. shortcuts

You can perform almost any action in darktable with a keyboard/mouse shortcut. You can also use various other input devices, including MIDI devices and game controllers – see the [midi device support](#) section for details. These are referred to as *external devices* or just *devices* in this guide.

## defining shortcuts

A *shortcut* is a combination of key or button presses and/or mouse or device movements that performs an *action* in darktable.

A single action may have multiple shortcuts but a single shortcut can only be linked to one action in a given darktable view – you can't chain actions together except by applying a preset or style. You can, however, set up a single shortcut that does one thing in the lighttable view, say, and another in the darkroom view.

## initiating a shortcut

A shortcut must be initiated by either

- pressing a key on the keyboard; or
- pressing a button or moving a knob/joystick on an external device

You cannot initiate a shortcut by moving your mouse or pressing one of its buttons, as these actions are used to interact with darktable's UI.

---

## simple shortcuts

A shortcut that only includes button and/or key presses (and not mouse/device movements) is referred to as a *simple* shortcut.

A simple shortcut must be initiated as above, but can include:

- One or more modifier keys (Shift, Ctrl, Alt), held down while executing the remainder of the shortcut
- Up to three key presses, the last one of which may be a long-press (defined as a key-press longer than your system's double-click duration)
- Similarly, up to three device-button presses or mouse-button clicks, the last of which may be long

Various combinations of keyboard, mouse, and device buttons can be used to create simple shortcuts.

## creating additional modifiers

The only valid modifiers are the Shift, Ctrl and Alt keys on the keyboard. You can define additional keys (or device buttons) as modifiers by assigning keys/buttons to the the “global/modifier” action. However, these will merely function as extra Ctrl, Alt or Shift keys – you cannot create “new” modifiers.

## extending simple shortcuts with movement

For certain actions you can choose to *extend* a simple shortcut using mouse/device movement. For example you might hold Ctrl+X while scrolling with your mouse to change the value of a slider. The following can be used to extend a simple shortcut:

- Movement of the mouse scroll wheel
- Horizontal, vertical or diagonal movement of the mouse cursor
- Movement of a knob/joystick on an external device

To extend a simple shortcut, you must hold the final key/button of the simple shortcut while performing the extending mouse/device movement.

For external devices you do not need to start with a *simple* shortcut – you can directly assign a control knob or joystick to an action – though this will significantly reduce the flexibility of such devices.

Long button and key presses cannot be extended, as the length of the click/press is timed using the release of the final button/key – such shortcuts must be terminated with the raising of the final button/key.

---

**Note:** You may need to switch off the “disable touchpad while typing” setting if you want to use extended shortcuts with a laptop touchpad.

---

## actions

Shortcuts are used to initiate *actions* within darktable.

An *action* is usually (but not always) an operation that you might undertake using darktable's point-and-click user interface. For example:

- Increase, decrease or reset sliders
- Scroll through dropdown lists
- Enable, expand or focus modules
- Click buttons
- Switch between views

Such point-and-click type *actions* are normally defined as the application of an *effect* to an *element* of a *widget*, where these terms are defined as follows:

**widget**

Each visible part of the user interface is known as a *widget*. For example the darktable application window is a widget, containing side panel widgets, each of which contains module widgets, each of which contains button, slider and dropdown list widgets etc... When assigning a shortcut to an action, you must first decide which widget it is to be applied to.

**element**

An *element* is the part of a UI widget that is affected by your shortcut. For example, for a slider that has a color picker, you can make a shortcut activate the color picker *button* element or change the *value* element of the slider. For a row of tabs (the row is a single widget) you can select which tab element to activate.

**effect**

A shortcut can sometimes have multiple possible *effects* on a given *element*. For example, a button can be activated as if it was pressed with a plain mouse-click or as if it was pressed with Ctrl+click. A slider's value can be edited, increased/decreased or reset.






## assigning shortcuts to actions

There are two primary methods of assigning a shortcut to an action.

## visual shortcut mapping


Click on the  icon in the [top panel](#) of any darktable view to enter visual shortcut mapping mode.

The mouse cursor will change as you hover over UI widgets, to indicate whether or not a mapping can be created:

- A down arrow with a line  appears when you hover over a module header, to indicate that you can click to expand the module.
- A spiral  indicates that a shortcut can be defined for the widget under the cursor.
- An up arrow  indicates that, in addition to assigning a shortcut, you can also add the widget to the [quick access panel](#) in the darkroom (by Ctrl+clicking on it).
- A down arrow  indicates that the widget is already in the quick access panel (Ctrl+click to remove it).
- A cross  indicates that there is no mappable widget under the cursor.

Press a key combination while hovering over a mappable widget to assign a shortcut to that widget – a default action will be assigned to that shortcut based on the type of widget and whether you have keyed a *simple* or *extended* shortcut. See below for details of some of the default assigned actions.

Left-click on a mappable widget to open the shortcut mapping screen for that widget (see below). Left-click anywhere else on the screen to open the shortcut mapping screen, expanded (where possible) based on the part of the screen you have clicked on. This screen can be used to alter the action assigned to a shortcut and to configure shortcuts for non-visual actions. Entering the shortcut mapping screen exits visual shortcut mapping mode.

You can assign as many shortcuts as you like in a single mapping session and then exit mapping mode when you are finished by clicking the  icon again or right-clicking anywhere on the screen.

You can delete a user-defined shortcut mapping by defining it a second time against the same widget. If you attempt to reallocate an existing shortcut to a new action, you will be notified of the conflict and asked whether you wish to replace the existing shortcut.

## shortcut mapping screen

The most flexible way to create shortcuts is by using the shortcut mapping screen, which can be accessed from the global preferences dialog or by left-clicking in visual mapping mode. This screen allows access to all available actions, including some that are not directly linked to a UI widget.

The top panel of the shortcut mapping screen shows a list of available UI widgets/actions and the bottom panel shows the shortcuts currently assigned to them. You can search the top and bottom panels using the text entry boxes at the bottom of the screen.

Double-click an item in the top panel to create a new shortcut for that item, and then enter your desired shortcut (right-click to cancel). Once you have done this, a new entry will appear in the bottom panel showing the shortcut you have created. You can then manually alter the *element*, *effect*, *speed* or *instance* of the assigned action against that shortcut in the bottom panel. To delete a user-defined shortcut, select it in the bottom panel and press the Delete key.

The following additional options are provided in the shortcut mapping screen:

### export...

Export the current shortcut mappings for one or all of your devices (keyboard/mouse, midi, game) to an external file. The dialog will show you how many shortcuts exist for each device.

### import...

Import shortcut mappings from an external file for one or all of your devices. When loading a device, you can choose to assign it a different number. This can for example be used to exchange midi layouts. Before loading, you can choose to wipe the specific device first. When loading all from an empty file, this will effectively delete all your shortcuts.

### restore...

Restore your shortcut mappings to (a) The mappings shipped with darktable by default, (b) The start of your current session, or (c) The point at which the shortcut mapping screen was last opened. When restoring, you can choose to leave any additional shortcuts that were added after the relevant checkpoint as they are, so that only changed shortcuts are restored to their previous meaning. Or you can choose to first clear all shortcuts and just load the restore point.

## deleting default shortcuts

It is not currently possible to delete system-defined default shortcuts – if you delete a default shortcut it will be automatically reinstated the next time you restart darktable.

As a workaround, you can *override* the default by assigning an identical shortcut to the “global/no-op” action (which does nothing). You can do this either in the shortcut mapping screen (above) or by directly editing your `$HOME/.config/darktable/shortcutsrc` file. If you want to disable a lot of default shortcuts the latter option is recommended (you must exit darktable first). For example, the following default shortcuts are defined in `shortcutsrc` for switching views in darktable:

```
d=global/switch views/darkroom
l=global/switch views/lighttable
m=global/switch views/map
p=global/switch views/print
s=global/switch views/slideshow
t=global/switch views/tethering
```

You can disable all of these shortcuts by changing `shortcutsrc` as follows:

```
d=global/no-op
l=global/no-op
m=global/no-op
p=global/no-op
s=global/no-op
t=global/no-op
```

## common actions

The following is a list of some of the actions to which you can assign shortcuts, organized by widget type. This is not an exhaustive list and you are encouraged to browse the shortcut mapping screen for a complete list of available actions. If you assign a shortcut to a widget, it will be given a default action, depending on the type of widget and on whether you have assigned a simple or extended shortcut.

Note that it is possible to assign a number of actions that have no effect. For example, all sliders include a *button* element, regardless of whether such a button is actually present alongside a given slider.

## global

Actions in the “global” section of the shortcut mapping screen can be executed from any darktable view. Most of these actions do not have specific *elements* as they are used to perform one-off operations.

## views

Actions in the “views” section can only be executed from the specified darktable view. As with global actions, most do not have specific *elements* as they are used to perform one-off operations.

## buttons

A button is a clickable icon in the darktable interface. The default action, when assigning a simple shortcut to a *button*, is to activate that button as if clicked with the left mouse button. You can modify this action to activate the button as if clicked while holding a modifier key.

## toggles

A toggle is a button that has a persistent on/off state. It therefore has additional *effects* to allow you to toggle it or explicitly set its state. As with a normal button the default action, when assigning a simple shortcut to a toggle, is to activate the toggle as if clicked with the left mouse button (which toggles the button on/off).

## utility modules

All utility modules have the following elements:

### **show**

Acts as a *toggle* that expands and collapses the module.

### **reset**

Acts as a *button* that resets all module parameters when activated. The *ctrl-activate* action can be used to re-apply any automatic presets for that module.

### **presets**

Allows you to select actions from the [presets](#) menu (e.g. edit, update, previous, next). The default action, when assigning a simple shortcut to a *preset* element, is to display a list of the available presets for selection. Extended shortcuts are not currently available for preset elements.

The default action, when assigning a simple shortcut to a utility module, is to *toggle* the *show* element (expand/collapse the module).

In addition, shortcuts are available for all of the controls on each module as well as any stored presets (see below).

## processing modules

Processing modules have the same elements and defaults as utility modules with the following additional elements:

### **enable**

Acts as a *toggle* that switches the module on and off.

### **focus**

Acts as a *toggle* that focuses or defocuses the module. This is useful for modules such as [crop](#) or [tone equalizer](#) , whose on-screen controls are only activated when those modules have focus. For *crop*, changes are saved only when the module loses focus.

### **instance**

Allows you to select actions from the [multiple-instance](#) menu (e.g. move up/down, create new instance). The default action, when assigning a simple shortcut to the *instance* element, is to display a list of the available options for selection; An extended shortcut will move the *preferred module instance* (see below) up and down the pixelpipe.

If an action affects a processing module that can have multiple instances, you can choose which instance to adjust with a given shortcut. By default, all actions will affect the “preferred” instance, as defined using the settings in [preferences > miscellaneous > keyboard shortcuts with multiple instances](#) .

## dropdowns

A dropdown is a multi-selection box and has the following elements available:

### **selection**

Allows values to be selected from the dropdown list in various ways. The default action, when assigning a simple shortcut to a dropdown, is to display a popup *edit* box with a list of the available values for selection; An extended shortcut (including a mouse movement) will scroll through the available values.

### **button**

A standard *button* element that allows the button to the right of the dropdown (if present) to be activated. For example, the *aspect* dropdown in the [crop](#) module has a button that allows the crop controls to be changed from portrait to landscape and vice versa.

## sliders

A slider allows you to continuously alter an integer or decimal value, and has the following elements available:

### **value**

Allows the current value of the slider to be altered. The default action, when assigning a simple shortcut to a slider, is to display a popup *edit* box so you can enter a value; An extended shortcut (including a mouse movement) will change the value up and down. Value elements are also used for modifying some on-screen graphs. When modifying the *value* element with a shortcut you may not exceed the bounds set in the visual slider.

### **force**

This is the same as the *value* element described above, but it allows you to exceed the bounds set in the visual slider.

### **zoom**

Allows you to change the upper and lower bounds of the visual slider without altering the current value.

### **button**

A standard *button* element that allows the button to the right of the slider (if present) to be activated. For example, a slider may include a color picker to visually set its value based on selected elements of the image.

You can alter the value of a slider more quickly or slowly than normal by defining the *speed* of the action in the shortcut mapping screen. By default a *value* (or *force*) effect is given a speed of 1.0, which means that it is changed at the default rate defined by the given slider. You can alter the slider more quickly by increasing the speed (a speed of 10 makes the action 10x faster) or more slowly by decreasing it (a speed of 0.1 makes the action 10x slower).

## fallbacks

Where a *widget* can have multiple different *actions* applied to it, it can be tedious to set up individual shortcuts for each one of those actions. To make this process simpler, if you create a simple shortcut a number of effects can be made available by default as extensions to that shortcut. These are known as *fallbacks*.

While fallbacks are a powerful way to quickly set up multiple actions using predefined and consistent shortcuts, they will assign a lot of actions automatically (which might not be what you want), and can be hard to understand. Fallbacks are therefore disabled by default and you will need to click on the “enable fallbacks” check box in the shortcuts setup window to enable them.

To take a brief example, you could create a simple shortcut (e.g. Ctrl+R) against a processing module. This will automatically set up the following *fallback* effects using the defined shortcut, extended with mouse-clicks. In each case (except the first) you should hold the initial shortcut while clicking with your mouse. The final mouse-click will apply the action defined below:

- Ctrl+R (no mouse-click) to show/hide the module (the default fallback)
- Ctrl+R+left-click to enable/disable the module
- Ctrl+R+left-double-click to reset the module
- Ctrl+R+right-click to show the module’s preset menu
- Ctrl+R+right-double-click to show the module’s multiple instance menu

Similar fallbacks are defined for many common UI elements and all can be manually overridden.

Some fallback actions are defined using modifier keys (usually Ctrl+ and Shift+). In this case you must define an initial shortcut without such a modifier in order to be able to use these fallbacks. For example, if you assign Ctrl+R to an action, you cannot use a Ctrl+ fallback. Some default fallbacks of this type are provided for the *value* element and for horizontal/vertical movements in the (zoomed) central area - in this case, Shift+ increases the speed to 10.0 and Ctrl+ decreases the speed to 0.1.

To see a list of *all* of the default fallbacks, click the “enable fallbacks” checkbox in the shortcut mapping screen and select the “fallbacks” category in the top panel. To see the fallbacks for a given widget (e.g. a slider) just select that widget in the top panel. In both cases an additional item (also named “fallbacks”) will then appear in the bottom panel containing full details of the available fallbacks.

Fallbacks are only applied if no other shortcut using that combination has been explicitly created. In the above example, if you were to explicitly assign Ctrl+R+left-click to another action, the “enable/disable module” fallback would be ignored.

As with any other shortcut, fallback settings are fully customizable.

## 9.11. presets

This menu provides an overview of the [presets](#) that are defined for darktable’s modules, and allows you to modify some of their properties.

Pre-defined presets (those that are included by default within darktable) are shown with a lock symbol. Their properties cannot be changed.

User-defined presets can be imported from exported .dtpreset files using the “import” button at the bottom of the screen. You can export *all* user-defined presets to a single directory using the “export” button.

Delete a user-defined preset by selecting it and pressing the Delete key.

Edit a user-defined preset’s properties by selecting it and pressing Enter or double-clicking. This will open a dialog that allows the preset to be edited, exported to an external .dtpreset file or deleted.

See the [creating and editing presets](#) section for more information about the other properties that can be edited in this dialog.



## 9.12. lua options

### **lua scripts installer don't show again**

Check this box to hide the [lua scripts installer](#) in the lighttable if no lua scripts are installed.

# 10. Scripting with Lua

darktable comes with a versatile scripting interface language for functionality enhancement.

## 10.1. overview

Lua scripts can be used to define actions for darktable to perform when an event is triggered. One example might be calling an external application during file export in order to apply additional processing steps outside of darktable.

[Lua](#) is an independent project founded in 1993, providing a powerful, fast, lightweight, embeddable scripting language. Lua is widely used by many open source applications, in commercial programs, and for games programming.

darktable uses Lua version 5.4. Describing the principles and syntax of Lua is beyond the scope of this usermanual. For a detailed introduction see the [Lua reference manual](#).

The following sections will provide you with a brief introduction to how Lua scripts can be used within darktable.

## 10.2. basic principles: luarc files

At startup, darktable will automatically run the Lua scripts `$DARKTABLE/share/darktable/luarc` and `$HOME/.config/darktable/luarc` (where `$DARKTABLE` represents the darktable installation directory and `$HOME` represents your home directory).

This is the only time darktable will run Lua scripts by itself. These scripts can register callbacks to perform actions on various darktable events. This callback mechanism is the primary method of triggering lua actions.

## 10.3. a simple lua example

Let's start with a simple example that will print some code on the console. Create a file called `luarc` in darktable's configuration directory (usually `$HOME/.config/darktable/`) and add the following line to it:

```
print("Hello World !")
```

Start darktable and you will see the sentence "Hello World !" printed on the console. Nothing fancy but it's a start.

At this point, there is nothing darktable-specific in the script. We simply use Lua's standard `print` function to print a string. That's nice and all, but we can do better than that. To access the darktable API you first need to `require` it and save the returned object in a variable. Once this is done you can access the darktable API as subfields of the returned object. All of this is documented in darktable's [Lua API](#) reference manual.

```
local darktable = require "darktable"  
darktable.print_error("Hello World !")
```

Run the script and ... nothing happens. The function `darktable.print_error` is just like `print` but will only print the message if you have enabled lua traces by running darktable with "`darktable -d lua`" on the command line. This is the recommended way to do traces in a darktable lua script.

## 10.4. printing labeled images

The first example showed us the very basics of lua and allowed us to check that everything was working properly. Now let's do something a little bit more complex. Let's try to print a list of images that have a "red" label attached to them. But first of all, what is an image?

```

local darktable = require "darktable"
local debug = require "darktable.debug"
print(darktable.debug.dump(darktable.database[1]))

```

Running the code above will produce a lot of output. We will look at it in a moment, but first let's look at the code itself.

We know about `require darktable`. Here, we need to separately `require darktable.debug` which is an optional section of the API that provides helper functions to help debug lua scripts.

`darktable.database` is a table provided by the API that contains all images in the library database. Each entry in the database is an image object. Image objects are complex objects that allow you to manipulate your image in various ways (all documented in the `types_dt_lua_image_t` section of the API manual). To display our images, we use `darktable.debug.dump` which is a function that will take anything as its parameter and recursively dump its content. Since images are complex objects that indirectly reference other complex objects, the resulting output is huge. Below is a cut down example of the output.

```

toplevel (userdata,dt_lua_image_t) : /images/100.JPG
  publisher (string) : ""
  path (string) : "/images"
  move (function)
  exif_aperture (number) : 2.7999999523163
  rights (string) : ""
  make_group_leader (function)
  exif_crop (number) : 0
  duplicate_index (number) : 0
  is_raw (boolean) : false
  exif_iso (number) : 200
  is_ldr (boolean) : true
  rating (number) : 1
  description (string) : ""
  red (boolean) : false
  get_tags (function)
  duplicate (function)
  creator (string) : ""
  latitude (nil)
  blue (boolean) : false
  exif_datetime_taken (string) : "2014:04:27 14:10:27"
  exif_maker (string) : "Panasonic"
  drop_cache (function)
  title (string) : ""
  reset (function)
  create_style (function)
  apply_style (function)
  film (userdata,dt_lua_film_t) : /images
    1 (userdata,dt_lua_image_t): .toplevel
    [.....]
  exif_exposure (number) : 0.0062500000931323
  exif_lens (string) : ""
  detach_tag (function): toplevel.film.2.detach_tag
  exif_focal_length (number) : 4.5
  get_group_members (function): toplevel.film.2.get_group_members
  id (number) : 1
  group_with (function): toplevel.film.2.group_with
  delete (function): toplevel.film.2.delete
  purple (boolean) : false
  is_hdr (boolean) : false
  exif_model (string) : "DMC-FZ200"
  green (boolean) : false
  yellow (boolean) : false
  longitude (nil)
  filename (string) : "100.JPG"
  width (number) : 945
  attach_tag (function): toplevel.film.2.attach_tag

```

```

exif_focus_distance (number) : 0
height (number) : 648
local_copy (boolean) : false
copy (function): toplevel.film.2.copy
group_leader (userdata,dt_lua_image_t): .toplevel

```

As we can see, an image has a large number of fields that provide all sort of information about it. Here, we are interested in the “red” label. This field is a boolean, and the documentation tells us that it can be written. We now just need to find all images with that field and print them out:

```

darktable = require "darktable"
for _,v in ipairs(darktable.database) do
  if v.red then
    print(tostring(v))
  end
end
end

```

This code should be quite simple to understand at this point, but it contains a few interesting aspects of lua that are worth highlighting:

- `ipairs` is a standard lua function that will iterate through all numeric indices of a table. We use it here because `darktable`'s database has non-numeric indices which are functions to manipulate the database itself (adding or deleting images, for example).
- Iterating through a table will return both the key and the value used. It is conventional in lua to use a variable named “\_” to store values that we don't care about.
- Note that we use the standard lua function `tostring` here and not the `darktable`-specific `darktable.debug.dump`. The standard function will return a name for the object whereas the debug function will print the content. The debug function would be too verbose here. Once again, it is a great debug tool but it should not be used for anything else.

## 10.5. adding a simple shortcut

So far, all our scripts have done things during startup. This is of limited use and doesn't allow us to react to real user actions. To do more advanced things we need to register a function that will be called on a given event. The most common event to react to is a keyboard shortcut.

```

darktable = require "darktable"

local function hello_shortcut(event, shortcut)
darktable.print("Hello, I just received '"..event..'
  "' with parameter '"..shortcut.."'")
end

darktable.register_event("shortcut",hello_shortcut,
  "A shortcut that prints its parameters")

```

Now start `darktable`, go to “preferences > shortcuts > lua > A shortcut that prints its parameters”, assign a shortcut and try it. You should see a nice message printed on the screen.

Let's look at the code in detail. We first define a function that takes two strings as input parameters. The first one is the type of event triggered (“shortcut”) and the second is the name of the shortcut (“A shortcut that prints its parameters”). The function itself calls `darktable.print`, which will print the message as an overlay in the main window.

Once that function is defined, we register it as a shortcut callback. To do that we call `darktable.register_event` which is a generic function for all types of events. We tell it that we are registering a shortcut event, then we give the callback to call and finally, we give the string to use to describe the shortcut in the preferences window.

Let's try a shortcut that is a little more interactive. This one will look at the images the user is currently interested in (selected or moused-over) and increase their rating:

```
darktable = require "darktable"

darktable.register_event("shortcut",function(event,shortcut)
    local images = darktable.gui.action_images
    for _,v in pairs(images) do
        v.rating = v.rating + 1
    end
end,"Increase the rating of an image")
```

At this point, most of this code should be self explanatory. Just a couple of notes:

- Instead of declaring a function and referencing it, we declare it directly in the call to `darktable.register_event` this is strictly equivalent but slightly more compact.
- `image.rating` is a field that gives the star rating of any image (between 0 and 5 stars, -1 means rejected).
- `darktable.gui.action_images` is a table containing all the images of interest. `darktable` will act on selected images if any image is selected, and on the image under the mouse if no image is selected. This function makes it easy to follow `darktable`'s UI logic in lua.

If you select an image and press your shortcut a couple of times, it will work correctly at first but when you have reached five stars, `darktable` will start showing the following error on the console:

```
<![CDATA[
LUA ERROR : rating too high : 6
stack traceback:
  [C]: in ?
  [C]: in function '__newindex'
  ./configdir/luarc:10: in function <./configdir/luarc:7>
    LUA ERROR : rating too high : 6
]]>
```

This is lua's way of reporting errors. We have attempted to set a rating of 6 to an image, but a rating can only go as high as 5. It would be trivial to add a check, but let's go the complicated way and catch the error instead:

```
darktable.register_event("shortcut",function(event,shortcut)
    local images = darktable.gui.action_images
    for _,v in pairs(images) do
        result,message = pcall(function()
            v.rating = v.rating + 1
        end)
        if not result then
            darktable.print_error("could not increase rating of image "..
                tostring(v).. " : "..message)
        end
    end
end,"Increase the rating of an image")
```

`pcall` will run its first argument and catch any exception thrown by it. If there is no exception it will return `true` plus any result returned by the function. If there is an exception it will return `false` and the error message of the exception. We simply test these results and print them to the console.

## 10.6. exporting images with lua

So far we have learned to use lua to adapt `darktable` to our particular workflow. Let's look now at how to use lua to easily export images to an online service. If you are able to upload an image to a service via the command line then you can use lua to integrate this into `darktable`'s user interface.

In this next example we will use lua to export via `scp`. A new storage type will appear in `darktable`'s UI that will export images to a remote target via the copy mechanism in `ssh`.

```

darktable = require "darktable"

darktable.preferences.register("scp_export","export_path",
    "string","target SCP path",
    "Complete path to copy to. Can include user and hostname","")

darktable.register_storage("scp_export","Export via scp",
    function( storage, image, format, filename,
        number, total, high_quality, extra_data)
        if not darktable.control.execute("scp "..filename.." "..
            darktable.preferences.read("scp_export",
                "export_path","string")) then
            darktable.print_error("scp failed for "..tostring(image))
        end
    end)
end)

```

`darktable.preferences.register` will add a new preference to `darktable`'s preferences menu, `scp_export` and `export_path` allow us to uniquely identify our preference. These fields are reused when we read the value of the preference. The `string` field tells the lua engine that the preference is a string. It could also be an integer, a filename or any of the types detailed in the API manual relating to `types_lua_pref_type`. We then have the label for the preference in the preference menu, the tooltip when hovering over the value and a default value.

`darktable.register_storage` is the call that actually registers a new storage type. The first argument is a name for the storage type, the second is the label that will be displayed in the UI and last is a function to call on each image. This function has a lot of parameters, but `filename` is the only one we use in this example. It contains the name of a temporary file where the image was exported by `darktable`'s engine.

This code will work but it has a couple of limitations. This is just a simple example after all:

- We use preferences to configure the target path. It would be nicer to add an element to the export UI in `darktable`. We will detail how to do that in the next section.
- We do not check the returned value of `scp`. That command might fail, in particular if the user has not correctly set the preference.
- This script can't read input from the user. The remote `scp` must use password-less copy – `scp` can't be provided with a password easily, so we will leave it that way.
- There is no message displayed once the example is done, only the progress bar on the lower left side tells the user that the job is complete.
- We use `coroutine.yield` to call an external program. The normal `os.execute` would block other lua codes from happening.

## 10.7. building user interface elements

Our previous example was a bit limited. In particular the use of a preference for the export path wasn't very nice. We can do better than that by adding elements to the user interface in the export dialog.

UI elements are created via the `darktable_new_widget` function. This function takes a type of widget as a parameter and returns a new object corresponding to that widget. You can then set various fields in that widget to set its parameters. You will then use that object as a parameter to various functions that will add it to the `darktable` UI. The following simple example adds a lib in the `lighttable` view with a simple label:

```

local my_label = darktable.new_widget("label")
my_label.label = "Hello, world !"

dt.register_lib("test","test",false,{
    [dt.gui.views.lighttable] = {"DT_UI_CONTAINER_PANEL_LEFT_CENTER",20},
    },my_label)

```

There is a nice syntactic trick to make UI elements code easier to read and write. You can call these objects as functions with a table of key values as an argument. This allows the following example to work. It creates a container widget with two sub-widgets: a label and a text entry field.

```

local my_widget = darktable.new_widget("box"){
  orientation = "horizontal",
  darktable.new_widget("label"){ label = "here => " },
  darktable.new_widget("entry"){ tooltip = "please enter text here" }
}

```

Now that we know that, let's improve our script a bit.

```

darktable = require "darktable"

local scp_path = darktable.new_widget("entry"){
  tooltip="Complete path to copy to. Can include user and hostname",
  text = "",
  reset_callback = function(self) self.text = "" end
}

darktable.register_storage("scp_export","Export via scp",
  function( storage, image, format, filename,
    number, total, high_quality, extra_data)
    if not darktable.control.execute(scp "..filename.." "..
      scp_path.text
    ) then
      darktable.print_error("scp failed for "..tostring(image))
    end
  end,
  nil, --finalize
  nil, --supported
  nil, --initialize
  darktable.new_widget("box") {
    orientation ="horizontal",
    darktable.new_widget("label"){label = "target SCP PATH "},
    scp_path,
  })

```

## 10.8. sharing scripts

So far, all of our lua code has been in *luarc*. That's a good way to develop your script but not very practical for distribution. We need to make this into a proper lua module. To do that, we save the code in a separate file (`scp-storage.lua` in this case):

```

--[
SCP STORAGE
a simple storage to export images via scp

AUTHOR
Jérémy Rosen (jeremy.rosen@enst-bretagne.fr)

INSTALLATION
* copy this file in $CONFIGDIR/lua/ where CONFIGDIR
is your darktable configuration directory
* add the following line in the file $CONFIGDIR/luarc
  require "scp-storage"

USAGE
* select "Export via SCP" in the storage selection menu
* set the target directory
* export your images

LICENSE
GPLv2

]]
darktable = require "darktable"

```



```

darktable.configuration.check_version(...,{2,0,0})

local scp_path = darktable.new_widget("entry"){
  tooltip="Complete path to copy to. Can include user and hostname",
  text = "",
  reset_callback = function(self) self.text = "" end
}

darktable.register_storage("scp_export","Export via scp",
  function( storage, image, format, filename,
    number, total, high_quality, extra_data)
  if coroutine.yield("RUN_COMMAND","scp "..filename.." "..
    scp_path.text
  ) then
    darktable.print_error("scp failed for "..tostring(image))
  end
end,
nil, --finalize
nil, --supported
nil, --initialize
darktable.new_widget("box") {
  orientation ="horizontal",
  darktable.new_widget("label"){label = "target SCP PATH "},
  scp_path,
})

```

darktable will look for scripts (following the normal lua rules) in the standard directories plus `$CONFIGDIR/lua/*.lua`. So our script can be called by simply adding `require "scp-storage"` in the `luarc` file. A couple of extra notes...

- The function `darktable.configuration.check_version` will check compatibility for you. The `"..."` will turn into your script's name and `{2,0,0}` is the API version you have tested your script with. You can add multiple API versions if you update your script for multiple versions of darktable.
- Make sure to declare all your functions as `local` so as not to pollute the general namespace.
- Make sure you do not leave debug prints in your code – `darktable.print_error` in particular allows you to leave debug prints in your final code without disturbing the console.
- You are free to choose any license for your script but scripts that are uploaded on darktable's website need to be GPLv2.

Once you have filled all the fields, checked your code, you can upload it to our script page [here](#) .

## 10.9. calling lua from dbus

It is possible to send a lua command to darktable via its Dbus interface. The method `org.darktable.service.Remote.Lua` takes a single string parameter which is interpreted as a lua command. The command will be executed in the current lua context and should return either `nil` or a string. The result will be passed back as the result of the Dbus method.

If the Lua call results in an error, the Dbus method call will return an error `org.darktable.Error.LuaError` with the lua error message as the message attached to the Dbus error.

## 10.10. using darktable from a lua script

*Warning: This feature is very experimental. It is known that several elements don't yet work in library mode. Careful testing is highly recommended.*

The lua interface allows you to use darktable from any lua script. This will load darktable as a library and provide you with most of the lua API (darktable is configured headless, so the functions relating to the user interface are not available).

As an example, the following program will print the list of all images in your library:

```
#!/usr/bin/env lua
package = require "package"
package.cpath=package.cpath..";./lib/darktable/lib?.so"

dt = require("darktable")(
  "--library", "./library.db",
  "--datadir", "./share/darktable",
  "--moduledir", "./lib/darktable",
  "--configdir", "./configdir",
  "--cachedir", "cachedir",
  "--g-fatal-warnings")

require("darktable.debug")

for k,v in ipairs(dt.database) do
  print(tostring(v))
end
```

Note the third line that points to the location of the `libdarktable.so` file.

Also note that the call to `require` returns a function that can be called only once and allows you to set darktable's command line parameter. The `:memory:` parameter to `--library` is useful here if you don't want to work on your personal library.

## 10.11. lua API

darktable's Lua API is documented in its own manual with a detailed description of all data structures and functions. You can download the API manual from [here](#).

# 11. Guides & Tutorials

## 11.1. developing monochrome images

Photography has a long history of producing monochrome images, and many still enjoy this aspect of photography. While there are some specialized/modified cameras with a truly monochrome sensor, most still use a regular camera to capture a color image and transform it into a monochrome image during post-processing.

There are two main approaches to this conversion:

- A *physical approach*, where we attempt to simulate how a silver-based photographic film emulsion would react to the light captured at the scene.
- A *perceptual approach*, where we develop a color image and reduce the color saturation in a perceptual color space such as *CIE Lab*.

These approaches, and other monochrome-related features in darktable, are discussed in the following sections.

## importing and flagging images as monochrome

When importing an image, there are a number of properties that can be used to indicate that the image requires a monochrome treatment:

- If the image was taken using an achromatic camera, the image will be automatically flagged as monochrome.
- When you capture a scene from which you would like to produce a monochrome image, it can be helpful to put your camera into a “black & white” creative mode. This allows you to visualise what the scene would look like in monochrome through your camera’s liveview screen or electronic viewfinder. The camera will still capture the full color data in the raw file, but the embedded jpeg preview image will be monochrome. When you import such an image, darktable can automatically flag the image as monochrome based on the preview image.

Checking whether the preview is monochrome slows down the import process, so this is disabled by default. You can enable this check in [preferences > processing > detect monochrome previews](#)

- When processing a raw file, one of the first steps is to [demosaic](#) the image. If you set the *demosaicing method* to “passthrough (monochrome)”, this discards color information during the demosaicing process, and darktable will flag the image as monochrome.

**Note:** You should only use this for images taken on a camera where the color filter array has been removed.

- After you have imported the image, you can manually flag an image as monochrome (or not) using the *metadata* tab on the lighttable’s [selected images](#) module,

If any of the above methods result in an image being flagged as monochrome, darktable modules can use this information to present the user with some monochrome-specific module controls and/or apply special processing to the image.

The `darktable|mode|monochrome` tag will be automatically applied to any images flagged as monochrome and, if you have enabled any permanent overlay information on your lighttable thumbnails, such images will be marked with a visual indicator B&W next to the file type information. By automatically applying this tag and visual indicator, darktable makes it easy to set up filters to single out images for monochrome treatment, and to see at a glance which images in the current collection bear the *monochrome* tag.

# monochrome conversion

## physical approach

This approach tends to work with linear scene-referred data from the sensor, and attempts to mimic the response of a photographic film with a silver emulsion. It consists of three steps:

1. Map the color channels from the sensor into a single monochrome channel. Different types of monochrome photographic film have different levels of sensitivity to various wavelengths of light, and this can be simulated by giving the three color channels different weightings when combining them together into a single monochrome channel. The [color calibration](#) module allows the three channels to be mixed into a gray channel by varying amounts, and it includes a number of presets that are designed to emulate the characteristics of some well-known types of photographic film.
2. Apply a luminosity saturation curve. As a piece of photographic film is exposed to more intense light, its response will fall off as the silver emulsion becomes saturated. This saturation curve can be simulated within the [filmic rgb](#) module.
3. Developing a monochrome film in the darkroom traditionally involves “dodging and burning” to control the level of exposure across different parts of the image. This can be emulated in darktable by using either the [exposure](#) module with manually created [masks](#), or by using the [tone equalizer](#) module, which generates a mask using a guided filter.

## perceptual approach

The other option for producing a monochrome image is to reduce the color saturation in the image, which can be done in a linear colorspace, or in a color space oriented towards modelling human perception.

- The [color balance](#) module operates in linear RGB, and allows you to reduce the color saturation in the image using either the input or output color saturation slider – which you choose depends on whether you want to make any other adjustments to either the color or monochrome image in the color balance module. The color balance module tends to give a predictable and perceptually-uniform result.
- The [monochrome](#) module works in Lab color space, and it allows the user to graphically define a weighted combination of colors to determine the density of the blacks in the monochrome image. The interface can be somewhat sensitive to the settings, with small changes producing large effects, and you may experience problems with the global contrast and/or black pixel artifacts.
- Other modules such as [color zones](#) can also be used to remove color saturation from the image, but these don't offer any real advantage over the simplicity of the [color balance](#) module's saturation sliders.

## 11.2. other resources

For help and support with using darktable you are encouraged to ask questions on the main discussion forums at [discuss.pixls.us](https://discuss.pixls.us).

The official places to obtain information concerning darktable are the following:

- [darktable.org](https://darktable.org)
- [GitHub wiki](#)

The following articles provide some useful background information about darktable's scene-referred workflow:

- [darktable 3.0 for dummies in 3 modules](#)
- [darktable 3.0 for dummies - hardcore edition](#)
- [darktable 3.0: RGB or Lab? Which modules? Help!](#)
- [darktable's filmic FAQ](#)
- [Introducing color calibration module](#)

The following resources can provide a deeper understanding of the functionality of some specific processing modules:

- [Image processing and the pixel pipeline in darktable 3.0](#)
- [Filmic RGB v3: remapping any dynamic range in darktable 3.0](#)
- [Filmic RGB v4: highlights reconstruction in darktable 3.2](#)
- [Dodging and burning with tone equalizer in darktable 3.0](#)
- [Noise reduction \(profiled\) with non-local means in darktable 3.0](#)
- [Noise reduction \(profiled\) with wavelets in darktable 3.0](#)
- [Contrast equalizer modules in darktable](#)

Some other YouTube channels with recent content on using darktable are:

- [Bruce Williams Photography](#)
- [Boris Hajdukovic](#)

# 12. Special Topics

## 12.1. color management

### 12.1.1. overview

darktable employs a fully color managed workflow:

- Input color specifications are taken from embedded or user-supplied ICC profiles or (in the case of raw files) from a library of camera-specific color matrices.
- darktable automatically reads the display profile of your monitor (if properly configured) for accurate on-screen color rendition. Multiple-screen setups are fully supported as long as a system service like colord is in place and properly configured to inform darktable of the correct monitor profile.
- Output files can be encoded in one of darktable's built-in profiles, including sRGB and Adobe RGB, or into a color space supplied by the user as an ICC profile.

### 12.1.2. display profile

For darktable to faithfully render colors on screen it needs to find the correct display profile for your monitor. In general this requires your monitor to be properly calibrated and profiled, and it needs the profile to be correctly installed on your system. darktable queries your X display server's xatom as well as the system service colord (if available) for the right profile. If required you can enforce a specific method in [preferences > miscellaneous](#) .

To investigate your display profile configuration you can run the [darktable-cmstest](#) binary (Linux only) which prints out useful information (e.g. profile name per-monitor) and tells you if the system is correctly configured.

In rare cases you may need to manually select the display profile. This is possible from within the [soft proof](#) and [gamut check](#) option dialogs in the darkroom view and the display profile dialog in the lighttable view.

Bear in mind that high-tier consumer-grade screens usually don't need a user-made display profile unless you need to perform soft-proofing with professional expectations, since they are properly calibrated to sRGB in the factory.

A poorly-made display profile will be more harmful than sticking to the default sRGB profile, since the default might be slightly inaccurate but will at least be reliable. Advanced and professional users are advised to proceed with the production of custom display profiles only if they have the training to assess the quality of the resulting profile and understanding of the profiling options.

### 12.1.3. rendering method

darktable can render colors either with its own internal algorithms or by using the external library LittleCMS2. darktable's internal method is, by an order of magnitude, faster than the external one. The external option gives you a choice of the rendering intent and might offer a slightly higher accuracy in some cases.

You can change the default method in [preferences > processing > always use LittleCMS 2 to apply output color profile](#)

---

**Note:** If the given ICC is LUT-based or contains both, a LUT and a matrix, darktable will use LittleCMS2 to render the colors regardless of the configuration parameter's value.

---

### 12.1.4. rendering intent

If rendering with LittleCMS2 is activated (see [rendering method](#) ) you can define how to handle out-of-gamut colors when converting between color spaces. A selection box in the [export](#) , [output color profile](#) , and [soft proof](#) modules gives you a choice of the following rendering intents:

#### **perceptual**

Best suited to photographs as it maintains the relative position of colors. This is usually the best choice.

**relative colorimetric**

Out-of-gamut colors are converted to colors having the same lightness, but different saturation. Other colors remain unmodified.

**saturation**

Saturation is retained but lightness is slightly changed.

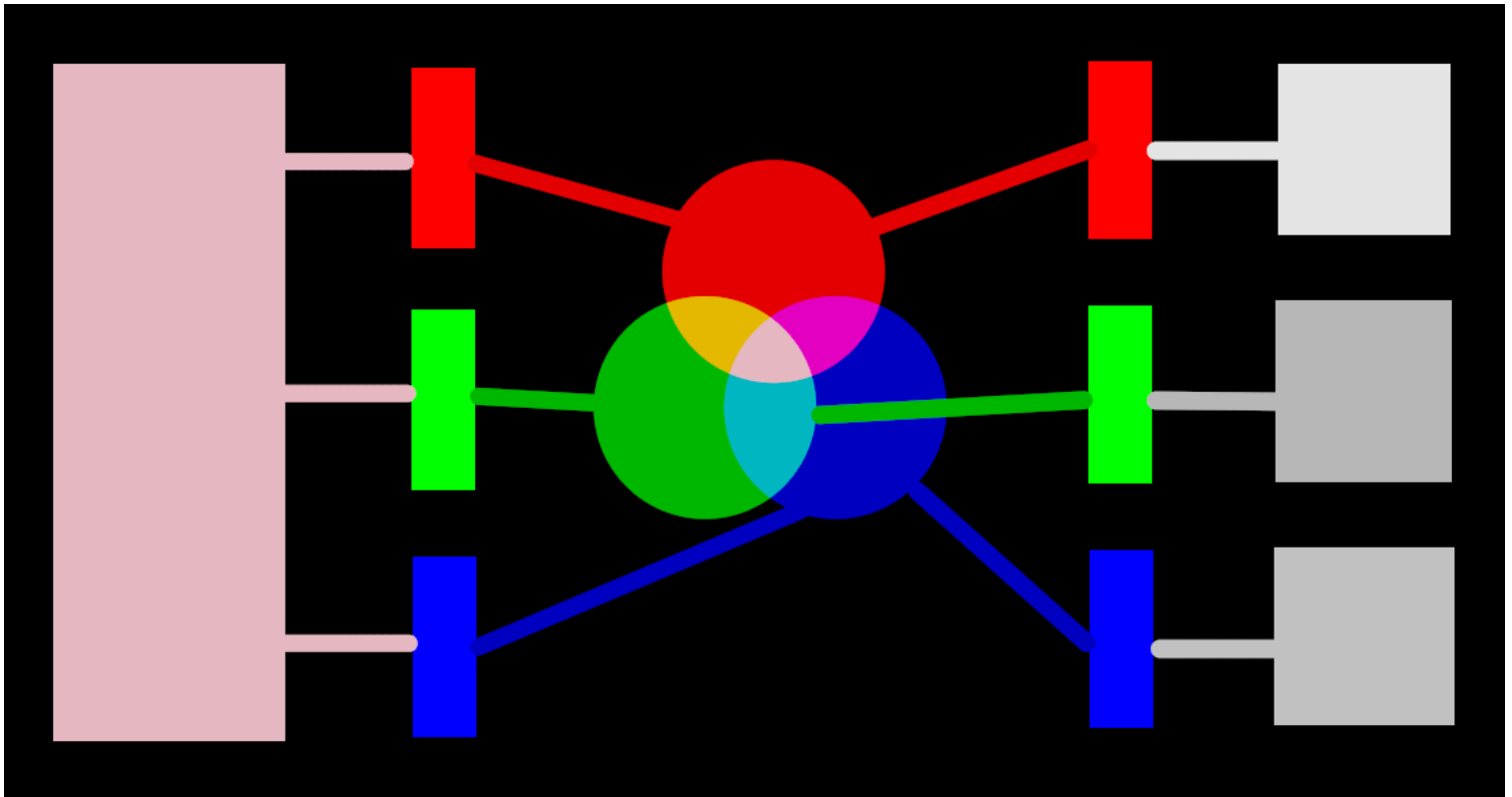
**absolute colorimetric**

Keep the white point.

## 12.1.5. darktable's color spaces

Input images are either RGB files (like JPEGs or TIFFs) or camera RAWs. Both store visual information as a combination of primary colors (e.g. red, green and blue) which together describe a light emission to be recreated by a display.

The following image illustrates this concept.



The left-hand-side of the image depicts a colored light that we need to represent digitally. We can use three ideal color filters to decompose this light into three colored primary lights at different intensities. In order to recreate the original colored light from our ideal decomposition (as illustrated in the center of the image) we simply need to recombine those three primary lights by addition.

It should be possible to reproduce the original colored light by taking a set of white lights at the correct intensities and projecting those lights through appropriately colored filters. This experiment can be performed at home using gels and dimmable white bulbs. This is roughly what old color CRT displays did and is how video projectors still work.

In photography, the initial decomposition step is performed by the color filter array that sits on top of your camera's sensor. This decomposition is not ideal, so it isn't possible to precisely recreate the original emission with simple addition – some intermediate scaling is required to adjust the three intensities.

On screens, the LED bulbs are dimmed proportionally to each intensity, and the emissions of the three lights are physically added to reconstruct the original emission. Digital images store the intensities of these primary lights as a set of three numbers for each pixel, depicted on the right-hand side of the above image as shades of gray.

While a set of display intensities can be easily combined to recreate an original light on a screen (for example, if we created a synthetic image in-computer) the set of captured intensities from a sensor needs some scaling in order for the on-screen light addition to reasonably reproduce the original light emission. This means that every set of intensities, expressed as an RGB set, must be linked to a set of filters (or primary LED colors) that define a *color space* – any RGB set only makes sense with reference to a color space.

Not only do we need to temper the captured intensities to make them summable again, but if we are to recompose the original light on a display that does not have the same colored filters or primaries as the space in which our RGB set belongs, these intensities need to be rescaled to take into account the differing filters on the display. The mechanism for this scaling is described in *color profiles*, usually stored within `.icc` files.

---

**Note:** Color is not a physical property of light – it exists only in the human brain, as a product of the decomposition of a light emission by the cone cells in the retina, again very similar in principle to the above filtering example. An “RGB” value should be understood as “light emissions encoded on 3 channels connected to 3 primaries”, but the primaries themselves may look different from what humans would call “red”, “green” or “blue”.

---

The filters described here are overlapping band-pass filters. Since they overlap, summing them back together would not preserve the energy of the original spectrum so (long story short) we need to dial them down with regard to the retina cone response

Most of darktable's actual image processing takes place in a large RGB “working profile” space, with some (mostly older) modules internally working in the CIE Lab 1976 color space (often just called “Lab”). The final output of the image processing pipeline is once again in an RGB space shaped for either the monitor display or the output file.

This process implies that the pixelpipe has two fixed color conversion steps: [input color profile](#) and [output color profile](#). In addition there is the [demosaic](#) step for raw images, where the colors of each pixel are reconstructed by interpolation.

Each module has a position in the pixelpipe that tells you which color space the module lives in:

- up to [demosaic](#) : The raw image information does not yet constitute an “image” but merely “data” about the light captured by the camera. Each pixel carries a single intensity for one primary color, and camera primaries are very different from primaries used in models of human vision. Bear in mind that some of the modules in this part of the pipe can also act on non-raw input images in RGB format (with full information on all three color channels).
- between [demosaic](#) and [input color profile](#) : Image is in RGB format within the color space of the specific camera or input file.
- between [input color profile](#) and [output color profile](#) : Image is in the color space defined by the selected working profile (linear Rec2020 RGB by default). As darktable processes images in 4x32-bit floating point buffers, we can handle large working color spaces without risking banding or tonal breaks.
- after [output color profile](#) : Image is in RGB format as defined by the selected display or output ICC profile.



## 12.1.6. unbounded colors

Screens and most image file formats can only encode RGB intensities confined within a certain range. For example, images encoded on 8 bits can only contain values from 0 to 255, images on 10 bits from 0 to 1023, and so on... Graphic standards postulate that the maximum of that range, no matter its actual value, will always represent the maximum brightness that the display medium is able to render, usually between 100 and 160 Cd/m<sup>2</sup> (or nits) depending on the actual standard. We generally call this maximum “100 % display-referred”. The minimum of the range, encoded 0 no matter the bit-depth used, becomes then “0 % display-referred”. 100 % encodes pure white, 0 % encodes pure black.

This is a limitation for image processing applications, because it means that any pixel lying outside of this range will be clipped to the nearest bound, resulting in non-recoverable loss of data (colors and/or textures).

For the longest time, image processing software too was bounded to this limitation for technical reasons, and some still is, but now by design choice. As a result, they would clip RGB intensities at 100 % display-referred between image operations.

darktable uses floating-point arithmetic inside its color pipeline, which means it can handle any RGB value internally, even those outside the display-referred range, as long as it is positive. Only at the very end of the pipeline, before the image is saved to a file or sent to display, are the RGB values clipped if needed.

Pixels that can take values outside of the display range are said to have “unbounded colors”. One could choose to clamp (i.e. confine) those values to the allowed range at every processing step or choose to carry on with them, and clamp them only at the last step in the pipeline. However, it has been found that processing is less prone to artifacts if the unbounded colors are not clamped but treated just like any other color data.

At the end of the pipeline, modules like [filmic](#) can help you to remap RGB values to the display-referred range while maximizing the data preservation and avoiding hard clipping, which is usually not visually pleasing.

However, at all times in the pipeline, you must ensure that you do not create negative RGB values. RGB intensities encode light emissions and negative light does not exist. Those modules that rely on a physical understanding of light to process pixels will fail if they encounter a non-physical light emission. For safety, negative RGB values are still clipped whenever they might make the algorithms fail, but the visual result might look degraded. Negative values can be produced when abusing the *black level* in [exposure](#) or the *offset* in [color balance](#) and care should be taken when using these modules.

## 12.1.7. possible color artifacts

There are some infrequent situations that still can lead to problematic results unless the user takes some action. Some modules in Lab color space, like [levels](#) and [monochrome](#), rely on the fact that the L channels carries all lightness information, with the a and b channels purely representing chroma and hue. Unbounded colors with negative L values are especially problematic to these modules and can lead to black pixel artifacts.

It has been found that highly saturated blue light sources in the image frame are likely candidates for pixels with negative L values. If you are engaged in stage photography you should pay close attention to such lights appearing in images.

In order to mitigate this issue the [input color profile](#) module has a gamut clipping option. This option is switched off by default but can be activated if artifacts are observed. Depending on the settings, colors will be confined to one of the available RGB gamuts. In effect black pixel artifacts are prevented at the costs of losing some color dynamics.

## 12.1.8. darktable's color dimensions

This section defines the perceptual properties of color, both conceptually and quantitatively, in order to characterize and quantify the creative and corrective adjustments made to color in darktable.

## definitions

Color properties like “saturation”, “brightness” or “lightness” have passed into common usage but are largely misused and often used to mean different things. In color science, each of these terms has a precise meaning.

There are two frameworks within which color properties may be analyzed and described:

- A scene-linear, physiological, framework, that mostly focuses on the response of the retina cone cells, using color spaces such as CIE XYZ 1931 or CIE LMS 2006,
- A perceptual, psychological, framework that stacks the brain’s corrections on top of the retina signal, using color spaces such as CIE Lab 1976, CIE Luv 1976, CIE CAM 2016 and JzAzBz (2017).

These two frameworks provide us with metrics and dimensions to analyze color and allow us to change some of its properties while preserving others.

The following dimensions of color are used by darktable:

### hue

An attribute of visual perception in which an area appears to be similar to one of the colors red, yellow, green, or blue, or to a combination of adjacent pairs of these colors considered in a closed ring. <sup>1</sup> Hue is a shared property between the perceptual and scene-linear frameworks.

### luminance

The density of luminous intensity with respect to a projected area in a specified direction at a specified point on a real or imaginary surface. <sup>2</sup> Luminance is a property of scene-referred frameworks, and is expressed by the Y channel of the CIE XYZ 1931 space.

### brightness

An attribute of visual perception according to which an area appears to emit, transmit or reflect more or less light. <sup>3</sup>

### lightness

The brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting. <sup>4</sup> Lightness is the perceptual, non-linear homologue of luminance (roughly equal to the cubic root of luminance Y). Lightness is expressed by the L channel in CIE Lab and Luv 1976 and the J channel in JzAzBz.

### chroma

The colorfulness of an area judged as a proportion of the brightness of a similarly illuminated area that appears gray, white or highly transmitting. <sup>5</sup> *Warning: chroma is not short for chrominance, which is the color part of a video signal (the Cb and Cr channels in YCbCr, for example).*

### brilliance

The brightness of an area judged relative to the brightness of its surroundings. <sup>6</sup>

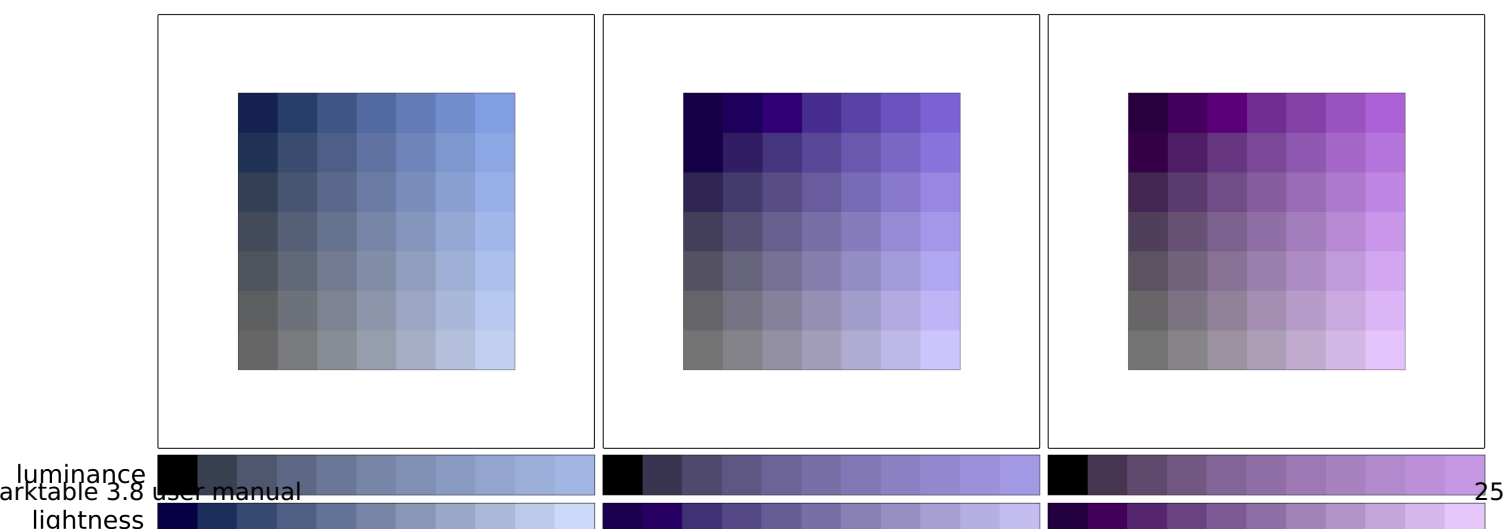
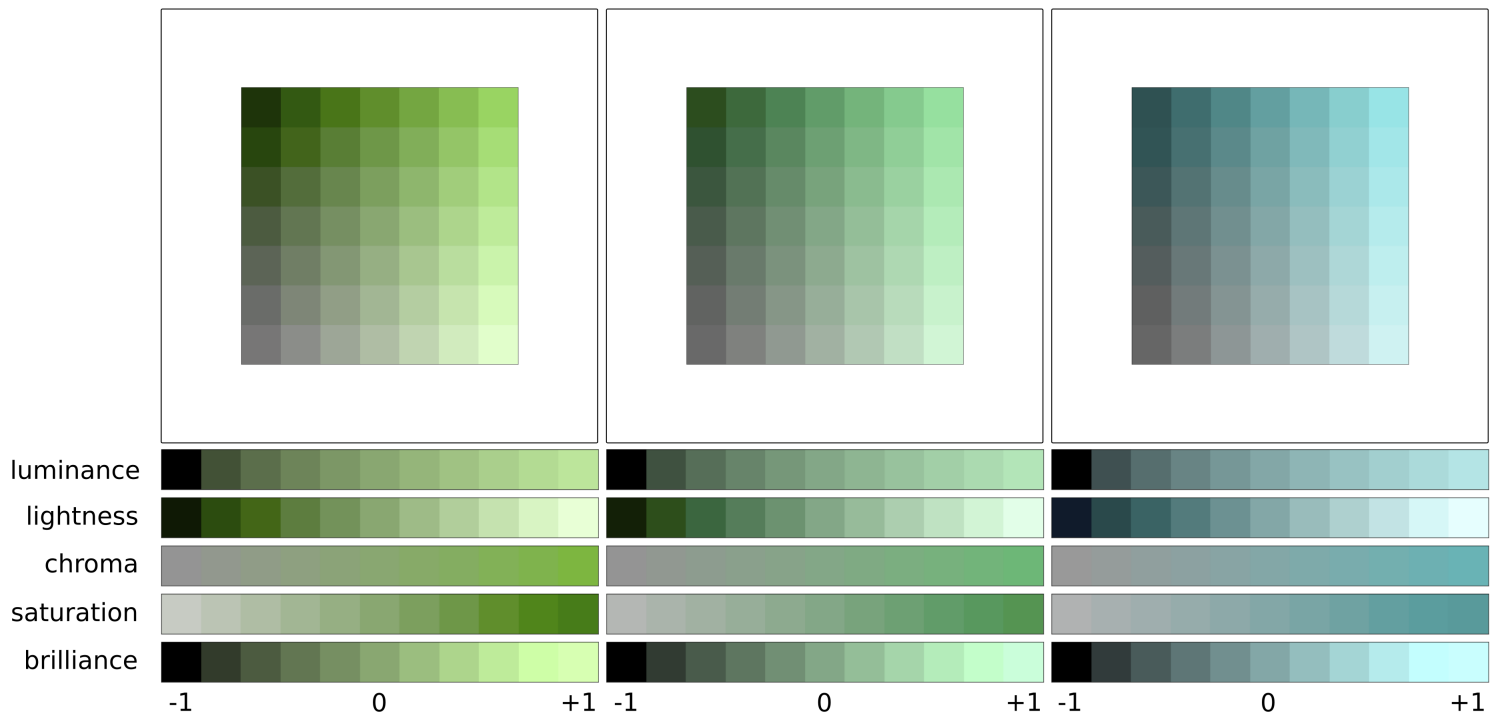
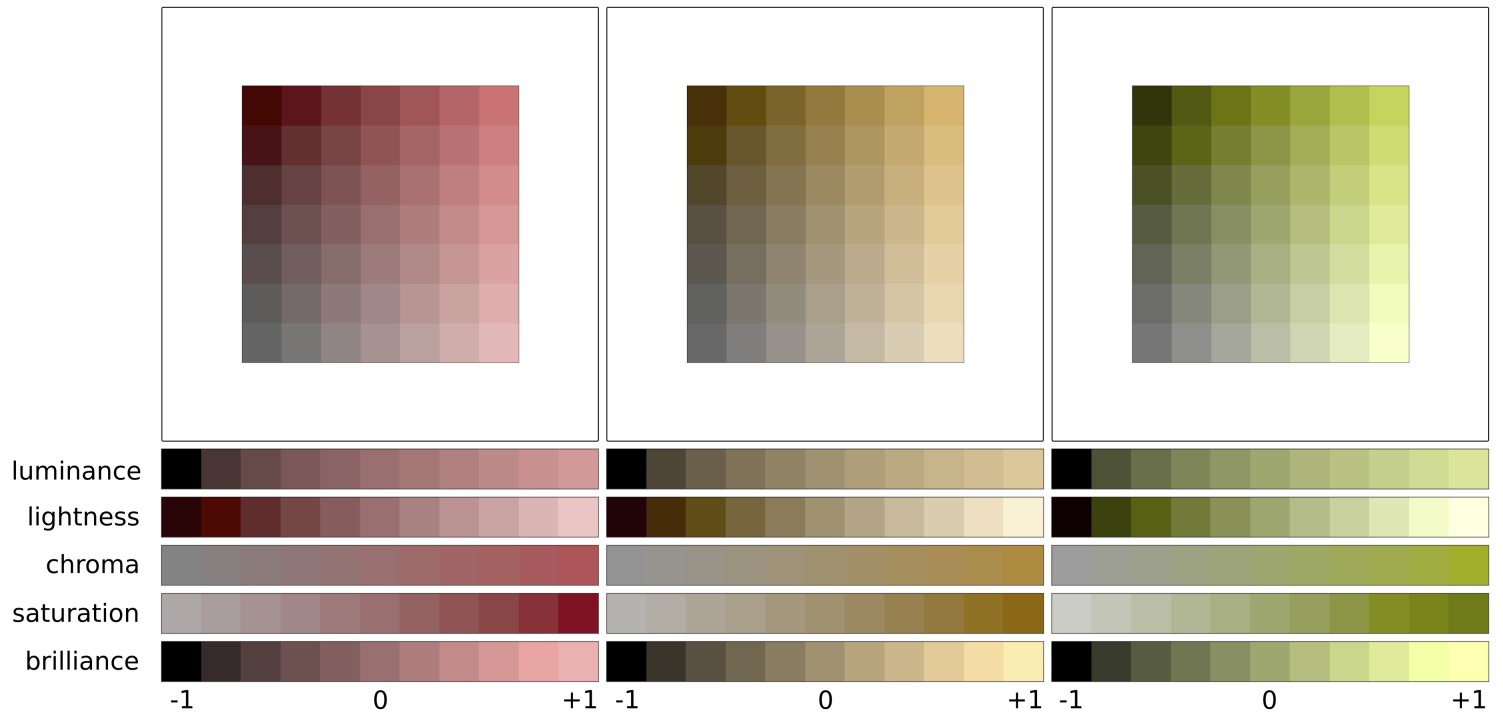
### saturation

The colorfulness of an area judged in proportion to its brightness. <sup>7</sup>

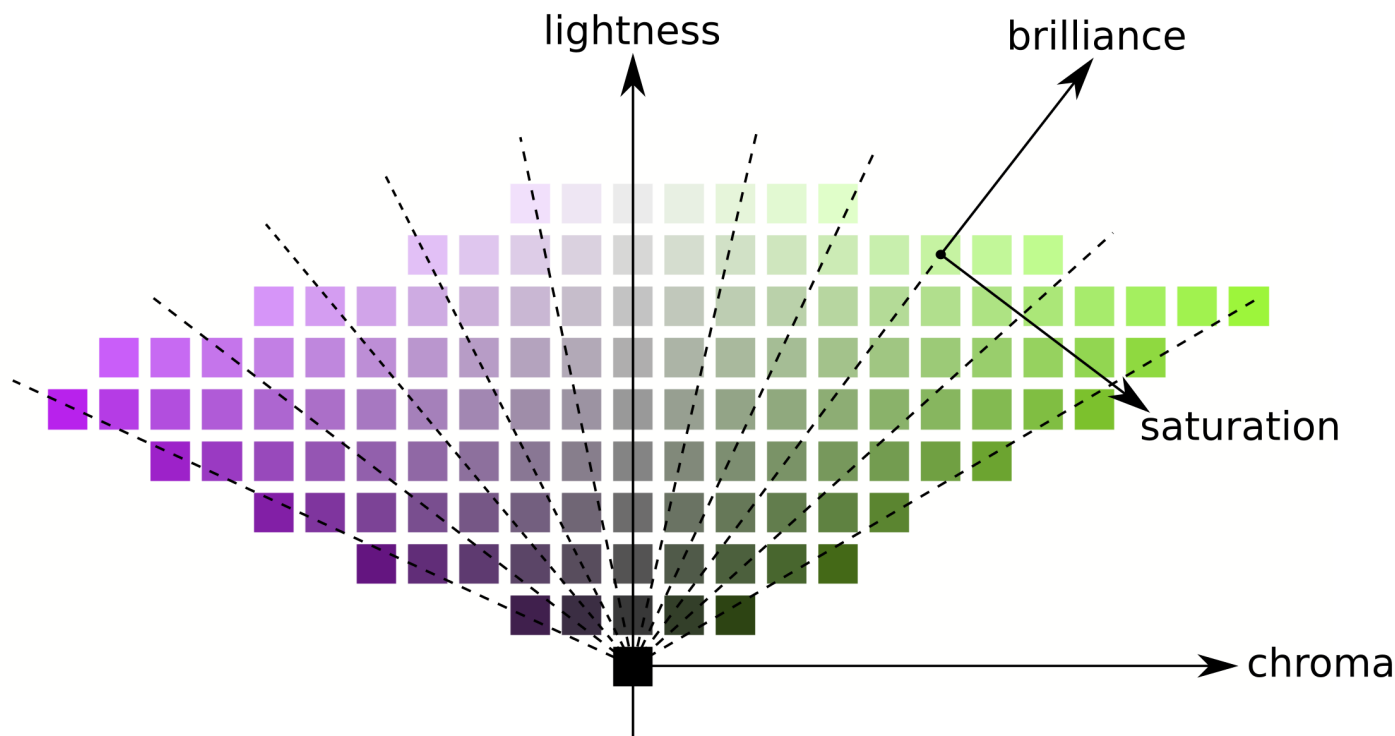
Colors can be described in many different color spaces, but no matter the color space, each color needs at least 3 components: some metric of luminance or brightness, and 2 metrics of chromaticity (hue and chroma, or opponent color coordinates).

## illustrations

While the previous definitions are useful to give a meaning to the words, they don’t tell us what we should be looking at. The following charts show luminance, lightness, chroma, brilliance/brightness and saturation varying from a “0” base color and how the resulting colors degrade:



(Lightness + Chroma) or (Brilliance/Brightness + Saturation) are two different ways to encode the same reality. They are orthogonal spaces that can be converted from one to another by a simple rotation of the base. This means that chroma evolves at constant lightness, saturation evolves at constant brilliance/brightness, and vice versa:



Lines of equal chroma are vertical (following the patches grid), meaning that chroma has the same direction for all colors in the gamut (see below). However, lines of equal saturation are oblique (drawn dashed on the graph) and all go from black through each color patch, meaning that their directions are particular to each color.

Increasing the chroma will therefore move all colors uniformly away from the central gray axis horizontally, while increasing the saturation will close or open the angle of the oblique dashed lines like a flower.

Similarly, increasing the lightness will move all colors uniformly up from the horizontal axis, while increasing brilliance/brightness will move them along the lines of equal saturation.

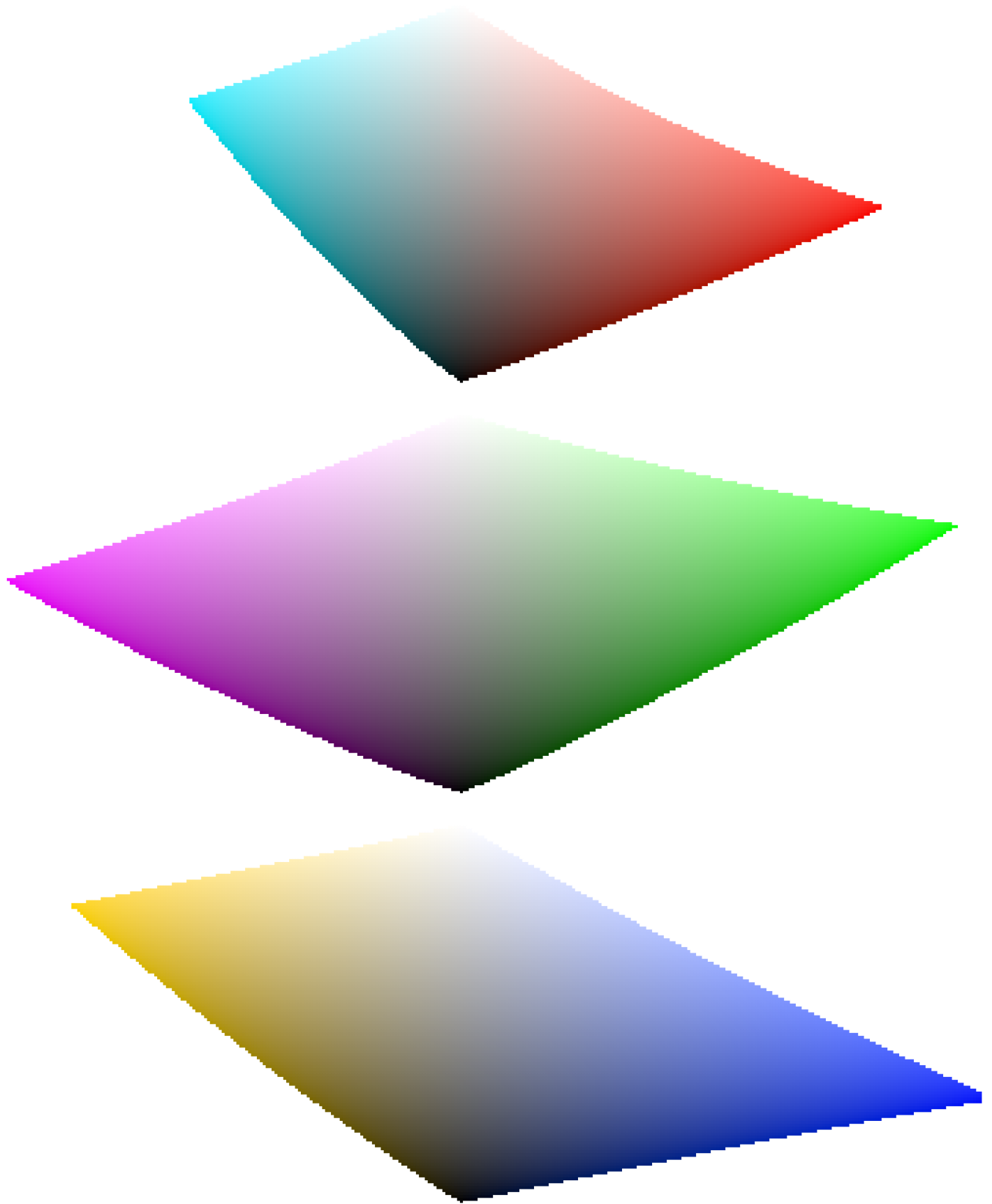
On both of the above charts, lightness, chroma, saturation and brilliance are drawn in the JzAzBz color space, which is a perceptual color space suited for an HDR signal, and is used in parametric masks and the color balance RGB module. Luminance is drawn in the CIE XYZ 1931 color space, and represents the effect of an exposure compensation. It shows the same behaviour as brilliance, except that the step size is not perceptually scaled.

**Note:** In this section, *brilliance* and *brightness* are both used to describe the same dimension. In all rigor, brightness is an absolute metric, whereas brilliance is the brightness of some surface relative to the brightness of its surroundings (that is, how much a surface “pops” out of its surroundings and looks fluorescent). But in image editing, increasing the brightness of some surface will indeed increase its brilliance too, so the term *brilliance* is preferred in darktable’s user interface for clarity and in reference to its visual effect.

## color dimensions and gamut

The gamut is the volume of colors that a certain color space can encompass and encode. It is important to note that, once converted to perceptual spaces, the gamut of any RGB space is not uniform along hues.

The following examples show the gamut volume of the sRGB space on hue slices containing the primary red, green and blue lights of the sRGB space, over a lightness-chroma plane with a uniform scale:

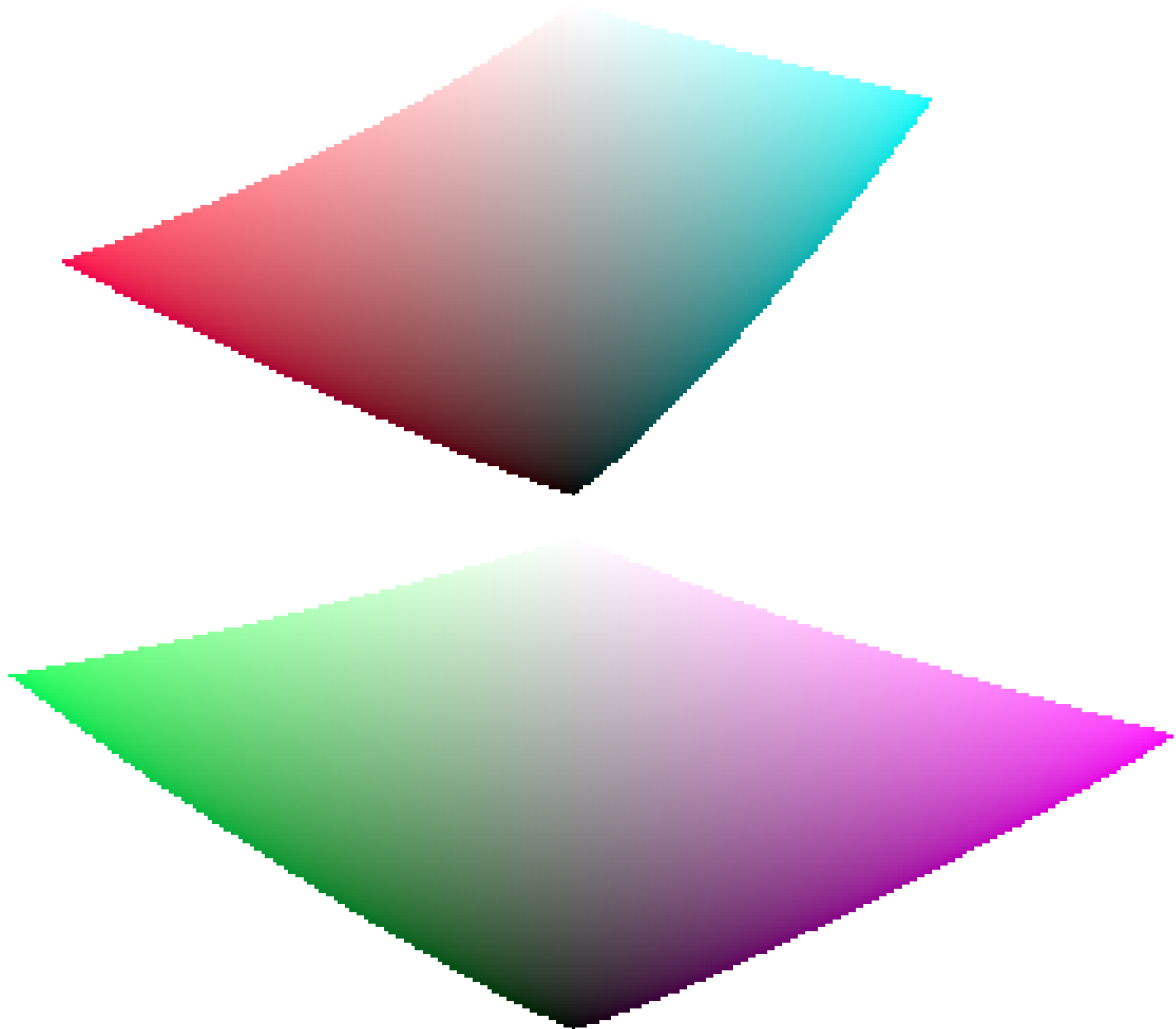


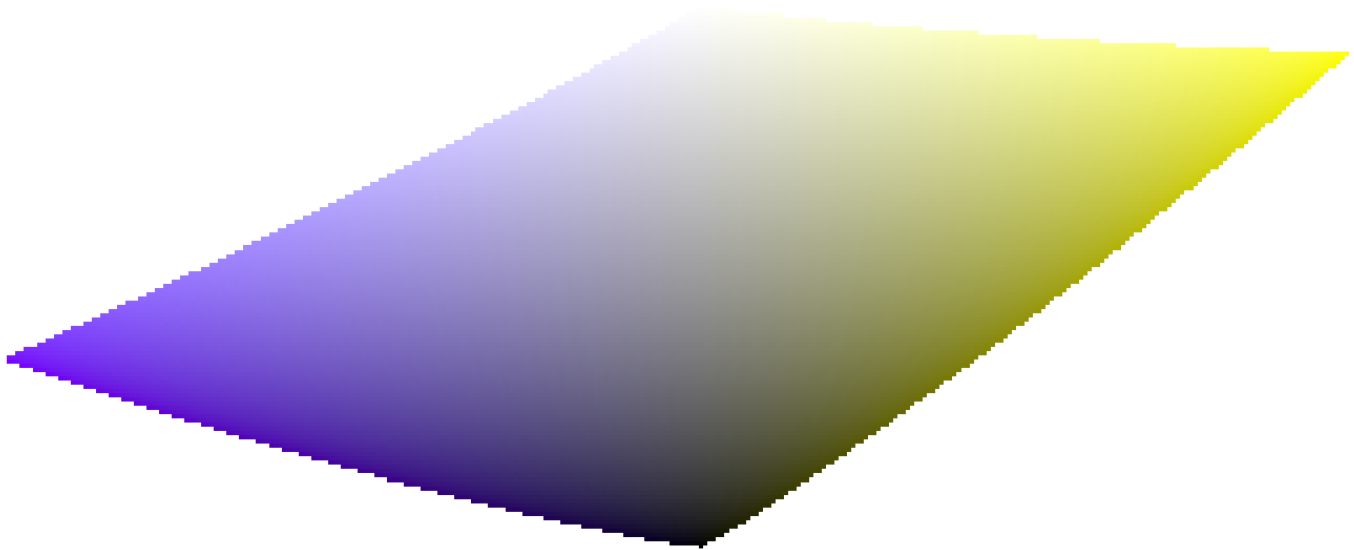
This shows that increasing the chroma (displacement over the horizontal axis) of some quantity can be safe for some hues at some lightness, but can push other hue-lightness coordinates way out of gamut. For example, we have much more margin in green or magenta than in cyan.

Many gamut issues at export are actually user-induced and the result of harsh chroma enlarging. For that reason, using brightness-saturation color models may be safer.

## color dimensions and complementary colors

Cyan, magenta, yellow (CMY) are complementary colors of red, green, blue (RGB). However, the complementary CMY spaces computed from RGB spaces are not perceptually complementary. To show this, we create a CMY space from sRGB, where cyan has sRGB coordinates (0, 1, 1), magenta (1, 0, 1) and yellow (1, 1, 0), and display it in a lightness-chroma space:





By comparing with the hue slices of the primary colors in the previous section, it is easy to see not only that the gamuts don't have the same shapes, but that the colors do not match.

This is one more reason to avoid using HSL/HSV spaces (derived from RGB spaces) to perform color editing: since these RGB spaces are not perceptually uniform in the first place, the resulting HSV/HSL spaces are not uniform either. While RGB spaces have some merit based on their connection to physical light, any process involving hue should go directly to perceptual spaces.

## color dimensions and settings

Many applications, including darktable, call any settings that affect chroma “saturation” (for example, in color balance, “contrast/brightness/saturation”). This is a symptom of software trying to be accessible to non-professionals by using a common language. This is misleading, since saturation does exist and is quite different from chroma. In addition, many video specifications improperly call chroma “saturation”. Whenever darktable reuses such specifications, it uses the incorrect term from the specification rather than the proper color dimension term.

1. CIE definition of *hue*: <https://cie.co.at/eilvterm/17-22-067> ↵
2. CIE definition of *luminance*: <https://cie.co.at/eilvterm/17-21-050> ↵
3. CIE definition of *lightness*: <https://cie.co.at/eilvterm/17-22-059> ↵
4. CIE definition of *brightness*: <https://cie.co.at/eilvterm/17-22-063> ↵
5. CIE definition of *chroma*: <https://cie.co.at/eilvterm/17-22-074> ↵
6. Article about *brilliance* (paywall): <https://doi.org/10.1002/col.20128> ↵
7. CIE definition of *saturation*: <https://cie.co.at/eilvterm/17-22-073> ↵

## 12.2. memory

darktable's memory requirements are high. A simple calculation makes this clear. If you have a 20MPx image then, for precision reasons, darktable will store this internally as a 4 x 32-bit floating point cell for each pixel. Each full image of this size will therefore need about 300MB of memory. As we want to process the image, we will at least need two buffers for each module - one for input and one for output. If we have a more complex module, its algorithm might additionally require several intermediate buffers of the same size. Without further optimization, anything between 600MB and 3GB would be needed merely to store and process image data. On top of that we have darktable's code segment, the code and data of all dynamically linked system libraries, and not to forget, further buffers where darktable stores intermediate images for quick access during interactive work (mip map cache).

All in all, darktable needs a minimum of about 4GB of memory to run happily.

## total system memory

From the above analysis, it is evident that your computer needs a sane memory setup to properly run darktable. We suggest that you have a least 4GB of physical RAM plus 4 to 8GB of additional swap space installed.

Theoretically, you could run darktable with lower amounts of physical RAM and balance this with enough swap space. However, you this could cause your system to “thrash”, as it reads or writes data pages to and from the hard disk. We have positive reports that this functions well for several users, but it still might get extremely slow for others. A solid-state drive can ease the pain slightly.

## available address space

Besides the total amount of system memory there is another limiting factor: the available address space of your hardware architecture. How much memory can be addressed by a process depends on the number of address bits your CPU offers. For a CPU with 32-bit address registers, this is  $2^{32}$  bytes, which makes a total of 4GB. This is the absolute upper limit of memory that can be used by a process and it constitutes a tight situation for darktable as we have seen above.

darktable’s escape route is called tiling. Instead of processing an image in one big chunk, darktable splits the image into smaller parts for every processing module. This still requires one full input and output buffer, but intermediate buffers can be made small enough to fit everything into the hardware’s limits.

## memory fragmentation

Unfortunately this is not yet the full story. An effect called memory fragmentation can and will hit software that needs to perform extensive memory management. If such a program allocates 5 times 300MB at a time and frees it again, that memory should normally be available for one big 1.5GB allocation afterwards. However this is often not the case. The system’s memory allocator may no longer see this area as one contiguous 1.5GB block but as a row of separate 300MB areas. If there is no other free area of 1.5GB available, a subsequent allocation would fail. During a program run this mechanism will take away more and more of the larger memory blocks in favor of smaller ones. darktable’s mip map cache allocates several small memory blocks per thumbnail, so this problem is even bigger. For this reason, as of darktable 2.0, 32-bit support is soft-deprecated.

## further limitations

As if this were not challenging enough, there are further things that might limit darktable’s access to memory. On some older boards you need to activate the “memory mapping” BIOS option in order to have all physically installed memory enabled. In addition if you are on a 32-bit OS you will probably need a kernel version that has “Physical Address Extension” (PAE) enabled. This is often but not always the case for Linux. Many distributions deliver different kernels, some with and some without PAE activated; you need to choose the right one. To check if the system is setup correctly, use the command “free” in a terminal and examine the output. If the output reports less RAM than you have installed, you have an issue needing correction; for example you have 4GB on your board, but your kernel is only seeing 3GB or less. You should consult your BIOS manual and the information about your Linux variant for further help.



## setting up darktable on 32-bit systems

As we've seen, 32-bit systems are difficult environments for darktable. Still some users are running darktable on them, if the basic requirements in terms of total system memory and the topics mentioned in the paragraphs above are addressed properly.

There are several parameters that require adjustment in order to get it running. If you install fresh, darktable will detect your system and set conservative values by default. However, if you upgrade darktable from an older version, chances are you have unfavorable settings in your preferences. The consequences might be darktable aborting due to allocation failures or – very typically – darktable not being able to properly import a new film roll. As a frequent symptom you get skulls displayed instead of thumbnails for many of your pictures.

If this is the case, take a minute to optimize your preference settings. You will find them in [preferences > processing > cpu/gpu/memory](#) .

Here is a short explanation of the relevant parameters and their proposed settings:

### **number of background threads**

This parameter defines the maximum number of threads that are allowed to run in parallel when importing film rolls or doing other background stuff. For obvious reasons on 32-bit systems you can only have one thread consuming resources at a time. So you need set this parameter to 1; anything higher will kill you.

### **host memory limit (in MB) for tiling**

This parameter tells darktable how much memory (in MB) it should assume is available to store image buffers during module operations. If an image can not be processed within these limits in one chunk, tiling will take over and process the image in several parts, one after the other. Set this to the lowest possible value of 500 as a starting point. You might experiment later whether you can increase it a bit in order to reduce the overhead of tiling.

### **minimum amount of memory (in MB) for a single buffer in tiling**

This is a second parameter that controls tiling. It sets a lower limit for the size of intermediate image buffers in megabytes. The parameter is needed to avoid excessive tiling in some cases (for some modules). Set this parameter to a low value of 8. You might tentatively increase it to 16 later.

### **memory in megabytes to use for thumbnail cache**

This controls how many thumbnails (or mip maps) can be stored in memory at a time. As a starting point set this to something like 256MB. Since darktable 2.0, the cache does allocate a few small buffers per thumbnail in cache, thus causing significant memory fragmentation. As explained before, this poses a problem for 32-bit systems. For this reason, as of darktable 2.0, 32-bit support is soft-deprecated.

## darktable on 64-bit systems

There's not much to be said here. Of course 64-bit systems also require a sufficient amount of main memory, so the 4GB plus swap recommendation holds true. On the other hand, 64-bit architectures do not suffer from the specific 32-bit limitations like small address space and fragmentation madness.

Most modern Intel or AMD 64-bit CPUs will have available address space in the range of several Terabytes. The word "modern" is relative in this context: all AMD and Intel CPUs introduced since 2003 and 2004, respectively, offer a 64-bit mode. Linux 64-bit has been available for many years.

All relevant Linux distributions give you the choice to install a 32-bit or a 64-bit version with no added cost. You can even run old 32-bit binaries on a 64-bit Linux. The only thing you need to do is invest some time into the migration. In the end we strongly recommend moving to a 64-bit version of Linux. There really is no reason not to.

On a 64-bit system you can safely leave the tiling related configuration parameters at their defaults: "host memory limit (in MB) for tiling" should have a value of 1500 and "minimum amount of memory (in MB) for a single buffer in tiling" should be set to 16. If you are migrating from a 32-bit to a 64-bit system you will need to check these settings and manually change them if needed in darktable's preference dialog.

Typically there is no need to restrict oneself in the number of background threads on a 64-bit system. On a multi-processor system a number of two to eight threads can speed up thumbnail generation considerably versus only one thread. The reason is not so much taking maximum advantage of all your CPU cores – darktable's pixelpipe uses all of them in parallel anyway – but hiding I/O latency.

One exception is worth mentioning. If you use darktable to process stitched panoramas (e.g. TIFFs as generated by Hugin) these images can reach considerable sizes. Each background thread needs to allocate enough memory to keep one full image plus intermediates and output in its buffers. This may quickly cause even a well equipped 64-bit system to run out of memory. In that case lower the number of background threads to only one.

## 12.3. openc1

### 12.3.1. the background

Processing high resolution images is a demanding task requiring a modern computer. In terms of both memory and CPU power, getting the best out of a typical 15, 20 or 25 Megapixel image can quickly take your computer to its limits.

darktable's requirements are no exception. All calculations are performed on 4 x 32bit floating point numbers. This is slower than "ordinary" 8 or 16 bit integer algebra, but eliminates all problems of tonal breaks or loss of information.

A great deal of optimization has been undertaken to make darktable as fast as possible. If you run a current version of darktable on a modern computer, you might not notice any "slowness". However, there are conditions and certain modules where you will feel (or hear from the howling of your CPU fan) how much your poor multi-core processor has to struggle.

That's where OpenCL comes in. OpenCL allows darktable to take advantage of the enormous power of modern graphics cards. Gamers' demand for highly detailed 3D worlds in modern shooters (as well as cryptocurrency mining) has fostered rapid GPU development. AMD, NVIDIA and Co had to put enormous processing power into their GPUs to meet these demands. The result is modern graphics cards with highly parallelized GPUs that can quickly calculate surfaces and textures at high frame rates.

You are not a gamer and you don't take advantage of that power? Well, then you should at least use it in darktable! For the task of highly parallel floating point calculations modern GPUs are much faster than CPUs. This is especially true when you want to repeat the same few processing steps millions of times. Typical use case: processing high megapixel images.

## 12.3.2. how openc1 works

As you can imagine, the hardware architecture of GPUs can vary significantly. There are different manufacturers, and even different generations of GPUs from the same manufacturer may not be comparable. At the same time GPU manufacturers don't normally disclose all the hardware details of their products to the public. One of the consequences of this is the need to use proprietary drivers under Linux, if you want to take full advantage of your graphics card.

Fortunately an industry consortium lead by The Khronos Group has developed an open, standardized interface called OpenCL, which allows your GPU to be used as a numerical processing device. OpenCL offers a C99-like programming language with a strong focus on parallel computing. An application that wants to use OpenCL will need OpenCL source code that it hands over to a hardware-specific compiler at run-time. This way applications can use OpenCL on different GPU architectures (even at the same time). All of the hardware "secrets" are hidden in this compiler and are normally not visible to the user (or the application). The compiled OpenCL code is loaded onto your GPU and - with certain API calls - it is ready to perform calculations for you.

## 12.3.3. activating openc1 in darktable

Using OpenCL in darktable requires that your PC is equipped with a suitable graphics card and that it has the required libraries in place. Most modern graphics cards from NVIDIA and AMD come with full OpenCL support. The OpenCL compiler is normally shipped as part of the proprietary graphics driver and is used as a dynamic library called `lib0penCL.so`. This library must be in a folder where it can be found by your system's dynamic linker.

When darktable starts, it will first try to find and load `lib0penCL.so` and, on success, check if the available graphics card comes with OpenCL support. A sufficient amount of graphics memory (1GB+) needs to be available for darktable to take advantage of the GPU. If that check passes, darktable tries to setup its OpenCL environment: a processing context needs to be initialized, a calculation pipeline to be started, OpenCL source code files (extension is `.cl`) needs to be read and compiled and the included routines (OpenCL kernels) need to be prepared for darktable's modules. If all of that completes successfully, the preparation is complete.

By default, OpenCL support is activated in darktable if all the above steps were successful. If you want to de-activate it you can do so in [preferences > processing > cpu/gpu/memory](#). This configuration parameter is grayed out if the OpenCL initialization failed.

You can switch OpenCL support off and on at any time without requiring a restart. Depending on the type of modules you are using, you will notice the effect as a general speed-up during interactive work and export. Most modules in darktable can take advantage of OpenCL but not all modules are demanding enough to make a noticeable difference. In order to feel a real difference, use modules like [shadows and highlights](#), [sharpen](#), [lowpass](#), [highpass](#) or even more extreme [contrast equalizer](#) and [denoise \(profiled\)](#).

If you are interested in profiling statistics, you can start darktable with command line parameters `-d openc1 -d perf`. After each run of the pixelpipe you will be shown details of processing time for each module plus an even more fine-grained profile for all used OpenCL kernels.

Apart from the speed-up you should not see any difference in the results between CPU and GPU processing. Except for some rounding errors, the results are designed to be identical. If, for some reason, darktable fails to properly finish a GPU calculation, it will normally detect the failure and automatically (and transparently) fall back to CPU processing.

## 12.3.4. setting up openc1

The huge diversity of systems and the marked differences between OpenCL vendors and driver versions makes it impossible to give an comprehensive overview of how to setup OpenCL. We only can give you an example, in this case for NVIDIA driver version 331.89 on Ubuntu 14.04. We hope that this will serve as a basic introduction and will help you to solve any problems specific to your setup.

The principle OpenCL function flow is like this:

```
darktable > libOpenCL.so > libnvidia-openc1.so.1 > kernel driver module(s) > GPU
```

- darktable dynamically loads `libOpenCL.so` – a system library that must be accessible to the system’s dynamic loader (`ld.so`).
- `libOpenCL.so` reads the vendor-specific information file (`/etc/OpenCL/vendors/nvidia.icd`) to find the library that contains the vendor-specific OpenCL implementation.
- The vendor-specific OpenCL implementation comes as a library `libnvidia-openc1.so.1` (which in our case is a symbolic link to `libnvidia-openc1.so.331.89`).
- `libnvidia-openc1.so.1` needs to talk to the vendor-specific kernel modules `nvidia` and `nvidia_uvm` via device special files `/dev/nvidia0`, `/dev/nvidiactl`, and `/dev/nvidia-uvm`.

At system startup the required device special files (`/dev/nvidia*`) need to be created. If this does not happen on your system by default, the easiest way to set them up and make sure all modules are loaded is by installing the `nvidia-modprobe` package.

A user account that needs to make use of OpenCL from within darktable must have read/write access to NVIDIA’s device special files. On some systems these files allow world read-write access by default, which avoids permission issues but might be debatable in terms of system security. Other systems restrict the access to a user group, e.g. “video”. In this case your user account has to be member of that group.

To summarise, the packages that needed to be installed in this specific case were:

```
nvidia-331 (331.89-0ubuntu1~xedgers14.04.2)
nvidia-331-dev (331.89-0ubuntu1~xedgers14.04.2)
nvidia-331-uvm (331.89-0ubuntu1~xedgers14.04.2)
nvidia-libopenc1-331 (331.89-0ubuntu1~xedgers14.04.2)
nvidia-modprobe (340.24-1)
nvidia-openc1-dev:amd64 (5.5.22-3ubuntu1)
nvidia-openc1-icd-331 (331.89-0ubuntu1~xedgers14.04.2)
nvidia-settings (340.24-0ubuntu1~xedgers14.04.1)
nvidia-settings-304 (340.24-0ubuntu1~xedgers14.04.1)
nvidia-libopenc1-331 (331.89-0ubuntu1~xedgers14.04.2)
nvidia-openc1-dev:amd64 (5.5.22-3ubuntu1)
nvidia-openc1-icd-331 (331.89-0ubuntu1~xedgers14.04.2)
openc1-headers (1.2-2013.10.23-1)
```

The list of NVIDIA related kernel modules as reported by `lsmod` was:

```
nvidia
nvidia_uvm
```

The list of NVIDIA related device special files (`ls -l /dev/nvidia*`) should read like:

```
crw-rw-rw- 1 root root 195,  0 Jul 28 21:13 /dev/nvidia0
crw-rw-rw- 1 root root 195, 255 Jul 28 21:13 /dev/nvidiactl
crw-rw-rw- 1 root root 250,  0 Jul 28 21:13 /dev/nvidia-uvm
```

Beware that the major/minor numbers (e.g. 250/0 for `/dev/nvidia-uvm` in this example) may vary depending on your system.

## 12.3.5. possible problems & solutions

darktable will detect OpenCL run-time errors automatically. On detecting an error, it will then reprocess everything on the CPU. While this will slow down processing it should not affect the end result.

There can be various reasons why OpenCL might fail during the initialization phase. OpenCL depends on hardware requirements and on the presence of certain drivers and libraries. In addition all these have to fit in terms of maker, model and revision number. If anything does not fit (e.g. your graphics driver – loaded as a kernel module – does not match the version of your `libOpenCL.so`) OpenCL support will likely not be available.

In this case, the best thing to do is start darktable from a console with `darktable -d openc1`.

This will give additional debugging output about the initialization and use of OpenCL. First, if you find a line that starts with `[openc1_init] FINALLY ...` that should tell you whether OpenCL support is available for you or not. If initialization failed, look at the messages above for anything that reads like `could not be detected` or `could not be created`. Check if there is a hint about where it failed.

Here are a few cases that have been observed in the past:

- darktable states that no OpenCL aware graphics card is detected or that the available memory on your GPU is too low and the device is discarded. In that case you might need to buy a new card if you really want OpenCL support.
- darktable finds your `libOpenCL.so` but then tell you that it couldn't get a platform. NVIDIA drivers will often give error code `-1001` in this case. This happens because `libOpenCL.so` is only a wrapper library. For the real work further vendor-specific libraries need to be loaded. This has failed for some reason. There is a structure of files in `/etc/OpenCL` on your system that `libOpenCL.so` consults to find these libraries. See if you can find something fishy in there and try to fix it. Often the required libraries cannot be found by your system's dynamic loader. Giving full path names might help.
- darktable states that a context could not be created. This often indicates a version mismatch between the loaded graphics driver and `libOpenCL`. Check if you have left-over kernel modules or graphics libraries from an older installation and take appropriate action. When in doubt, perform a clean reinstall of your graphics driver. Sometimes, immediately after a driver update, the loaded kernel driver does not match the newly installed libraries. In this case reboot your system before trying again.
- darktable crashes during startup. This can happen if your OpenCL setup is completely broken or if your driver/library contains a severe bug. If you can't fix it, you can still use darktable with option `--disable-openc1`, which will skip the entire OpenCL initialization step.
- darktable fails to compile its OpenCL source files at run-time. In this case you will see a number of error messages looking like typical compiler errors. This could indicate an incompatibility between your OpenCL implementation and darktable's interpretation of the standard. In that case please raise an issue on [github](#) and we will try to assist. Please also report if you see significant differences between CPU and GPU processing of an image.

A few on-CPU implementations of OpenCL also exist, coming as drivers provided by INTEL or AMD. We have observed that they do not provide any speed gain versus our hand-optimized CPU code. Therefore darktable simply discards these devices by default. This behavior can be changed by setting the configuration variable `openc1_use_cpu_devices` (in `$HOME/.config/darktable/erc`) to `TRUE`.

## 12.3.6. amd/ati devices

While NVIDIA devices and most modern AMD/ATI devices will often run out of the box, there is more to do for older AMD/ATI graphics cards, namely those prior to the HD7xxx series. This starts with the fact that those devices will only report part of their total GPU memory to darktable. For a 1GB device this typically amounts to only 512MB, a value that darktable in its standard configuration will refuse as insufficient. In consequence the device will not be used.

On the web you might find a tip to set environment variable `GPU_MAX_HEAP_SIZE` to 100 if this happens. Indeed this will cause the AMD/ATI driver to report the full installed memory to darktable. However, there is a problem. On many (most?) cards this will cause buffers to be allocated on your computer (host) and not on the video card! In this case all memory accesses will need to go through the slow PCIe bus. This will cost you a factor of 10x or more in performance and will render OpenCL useless for you, especially when exporting files.

Another environment variable that changes driver behavior is `GPU_MAX_ALLOC_PERCENT`. You could set this to 100 in order to allow memory allocations as high as 1GB on your AMD/ATI card. The problem is that this tends to cause darktable to crash sooner or later.

Our recommendation is therefore to leave these settings untouched. Often your card will be recognized with 512MB memory and a maximum allocation size of 128MB. There are three configuration parameters that you can set in `$HOME/.config/darktable/darktable.rc` to get things running. Here are the details:

### **openc1\_memory\_requirement**

Set this parameter to 500 so that darktable will accept your 512MB graphics memory as having sufficient memory.

### **openc1\_memory\_headroom**

This parameter controls how much graphics memory (from that reported by your card) darktable should leave untouched for driver and display use. Since for AMD/ATI devices we can only get half of the available RAM anyway, it's safe to set this to zero so that all of the 512MB can be used by darktable.

### **openc1\_avoid\_atomics**

Atomic operations in OpenCL are a special method of data synchronization. They are only used in a few kernels. Unfortunately, some (most?) AMD/ATI devices are extremely slow in processing atomics. It's better to process the affected modules on the CPU rather than accepting an ultra-slow GPU codepath. Therefore, set this parameter to TRUE if you experience slow processing of modules like [shadows and highlights](#), [monochrome](#), [local contrast](#), or [global tonemap \(deprecated\)](#) or if you get intermittent system freezes.

These recommendations do not apply to the more recent Radeon HD7xxx series with GCN architecture. Besides being very fast in terms of GPU computing they normally run out of the box, though you might consider trying some of the performance optimization options described in the following section.

## 12.3.7. performance optimization

There are a number of configuration parameters in `$HOME/.config/darktable/darktable.rc` that can help to finetune your system's OpenCL performance. Performance in this context mostly means the latency of darktable during interactive work (i.e. how long it takes to reprocess the pixelpipe). For a comfortable workflow it is essential to keep latency low.

In order to obtain profiling information you need to start darktable from a terminal with `darktable -d openc1 -d perf`.

After each reprocessing of the pixelpipe – caused by module parameter changes, zooming, panning, etc. – you will see the total time spent in the pixelpipe and the time spent in each of the OpenCL kernels. The most reliable value is the total time spent in the pixelpipe. Please note that the timings given for each individual module are unreliable when running the OpenCL pixelpipe asynchronously (see `openc1_async_pixelpipe` below).

To allow for fast pixelpipe processing with OpenCL it is essential that we keep the GPU busy. Any interrupts or a stalled data flow will add to the total processing time. This is especially important for the small image buffers we need to handle during interactive work. These can be processed quickly by a fast GPU. However, even short-term stalls of the pixelpipe can easily become a bottleneck.

On the other hand darktable's performance during file exports is more or less only governed by the speed of our algorithms and the horse-power of your GPU. Short-term stalls will not have a noticeable effect on the total time of an export.

darktable comes with default settings that should deliver a decent GPU performance on most systems. However, if you want to fiddle around a bit by yourself and try to optimize things further, here is a description of the relevant configuration parameters.

**openc1\_async\_pixelpipe**

This flag controls how often darktable blocks the OpenCL pixelpipe to get a status on success/failure of the kernels that have been run. For optimum latency set this to TRUE, so that darktable runs the pixelpipe asynchronously and tries to use as few interrupts as possible. If you experience OpenCL errors like failing kernels, set the parameter to FALSE. darktable will then interrupt after each module so that you can more easily isolate the problem. Problems have been reported with some older AMD/ATI cards, like the HD57xx, which can produce garbled output if this parameter is set to TRUE. If in doubt, leave it at its default of FALSE.

**openc1\_number\_event\_handles**

Event handles are used so that darktable can monitor the success/failure of kernels and profiling info even if the pixelpipe is executed asynchronously. The number of event handles is a limited resource of your OpenCL driver. For sure they can be recycled but there is a limited number that can be used at the same time. Unfortunately, there is no way to find out what the resource limits are, so darktable needs to guess. The default value of 25 is quite conservative. You might want to see if higher values like 100 give better OpenCL performance. If your driver runs out of free handles you will experience failing OpenCL kernels with error code -5 (CL\_OUT\_OF\_RESOURCES) or even crashes or system freezes. Reduce the number again if that happens. A value of 0 will block darktable from using any event handles. This will prevent darktable from properly monitoring the success of your OpenCL kernels but saves some driver overhead. The consequence is that any failures will likely lead to garbled output without darktable noticing. This is only recommended if you know for sure that your system runs rock-solid. You can also set this parameter to -1, which means that darktable assumes no restriction in the number of event handles. This is not recommended.

**openc1\_synch\_cache**

This parameter, if set to “true”, will force darktable to fetch image buffers from your GPU after each module and store them in its pixelpipe cache. This is a resource consuming operation, but can make sense depending on your GPU (including if the GPU is rather slow). In this case darktable might in fact save some time when module parameters have changed, as it can go back to some cached intermediate state and reprocess only part of the pixelpipe. In many cases this parameter should be set to “active module” (the default), which will only cache the input of the currently focused module.

**openc1\_micro\_nap**

In an ideal case you will keep your GPU busy at 100% when reprocessing the pixelpipe. That’s good. On the other hand your GPU may also be needed to do regular GUI updates. It might happen that there is no sufficient time left for this task. The consequence would be a jerky reaction of your GUI on panning, zooming or when moving sliders. To resolve this issue darktable can add small naps into its pixelpipe processing to have the GPU catch some breath and perform GUI related activities. The openc1\_micro\_nap parameter controls the duration of these naps in microseconds. You will need to experiment in order to find an optimum value for your system. Values of 0, 100, 500 and 1000 are good starting points to try. The default is 1000.

**openc1\_use\_pinned\_memory**

During tiling huge amounts of memory need to be transferred between host and device. On some devices (namely AMD) direct memory transfers to and from an arbitrary host memory region may give a huge performance penalty. This is especially noticeable when exporting large images. Setting this configuration parameter to TRUE tells darktable to use a special kind of intermediate buffer for host-device data transfers. On some devices this can speed up exporting of large files by a factor of 2 to 3. NVIDIA devices and drivers seem to have a more efficient memory transfer technique even for arbitrary memory regions. As they may not show any performance gain and even may produce garbled output, openc1\_use\_pinned\_memory should be left at its default FALSE for those devices.

**openc1\_building\_gpuXXX**

Manually add this setting to darktable.rc to add extra OpenCL compilation options for your GPU(s), where XXX is the GPU name. These options are used when compiling OpenCL kernels and may be provided for performance tuning or to work around bugs. You must remove any existing kernels in order to recompile them with the new options. Provide an empty string to recompile without any options. Remove the setting entirely to recompile with default options.

You can reference your GPU by its ID (e.g. openc1\_building\_gpu0) or by its canonical name (e.g. openc1\_building\_gpugeforce10606gb). Start darktable with darktable -d openc1 to find your canonical GPU name and default compilation options.

For example, the following lines would add extra compilation options for the GPU with ID 0 and for the GPU named “geforce10606gb”:

```
openc1_building_gpu0=-cl-mad-enable -cl-no-signed-zeros -cl-unsafe-math-optimizations -cl-finite-math-only -d
openc1_building_gpugeforce10606gb=-cl-mad-enable -cl-no-signed-zeros
```

## 12.3.8. scheduling profile

darktable can use the CPU and one or several OpenCL capable GPUs. Depending on the relative performance of these devices, users can choose among certain scheduling profiles to optimize performance. This is achieved by setting the configuration parameter [preferences > processing > cpu/gpu/memory > OpenCL scheduling profile](#) , which offers the following choices:

### **default**

If an OpenCL-capable GPU is found darktable uses it for processing the center image view while the [navigation preview window](#) is processed on the CPU in parallel. This is the preferred setting for systems with a reasonably fast CPU and a moderately fast GPU. The exact allocation of devices to the various pipeline types can be finetuned with the “openc1\_device\_priority” configuration parameter (see [multiple devices](#) ).

### **very fast GPU**

With this scheduling profile darktable processes the center image view and the preview window on the GPU sequentially. This is the preferred setting for systems with a GPU that strongly outperforms the CPU.

### **multiple GPUs**

This setting addresses systems with multiple GPUs whose relative performance does not differ significantly. Whenever a processing job is started darktable uses any currently idle GPU but not the CPU. Users of systems with a variety of GPUs will need better control on their relative priority. They would be better off selecting the “default” profile and fine-tuning their system with the “openc1\_device\_priority” configuration parameter (see [multiple devices](#) ).

On first start-up or after any detected change in the GPU configuration of your system darktable tries to identify the best suited profile for you. You can change it at any time in [preferences > processing > cpu/gpu/memory](#) .



## 12.3.9. multiple devices

The scheduling of OpenCL devices can be optimized on most systems using the “OpenCL scheduling profile” settings. However, if your system is equipped with more than one GPU, you might want to set the relative device priority manually. To do this you need to select the “default” scheduling profile and change the settings in the “openc1\_device\_priority” configuration parameter.

It is important to understand how darktable uses OpenCL devices. Each processing sequence of an image – to convert an input to the final output using a history stack – is run in a pixelpipe. There are four different types of pixelpipe in darktable. One type is responsible for processing the center image view (or full view) in darkroom mode, another pixelpipe processes the preview image (navigation window). There can be one of each of these two pixelpipes running at any time, with the full and preview pixelpipes running in parallel. In addition there can be multiple parallel pixelpipes performing file exports as well as multiple parallel pixelpipes generating thumbnails. If an OpenCL device is available, darktable dynamically allocates it to one specific pixelpipe for one run and releases it afterwards.

The computational demand varies significantly depending on the type of pixelpipe being executed. The preview image and thumbnails are low resolution and can be processed quickly, whereas processing the center image view is more demanding. A full export pixelpipe is more demanding still.

The configuration parameter “openc1\_device\_priority” holds a string with the following structure: `a,b,c.../k,l,m.../o,p,q.../x,y,z...`. Each letter represents one specific OpenCL device. There are four fields in the parameter string separated by a slash, each representing one type of pixelpipe. `a,b,c...` defines the devices that are allowed to process the center image (full) pixelpipe. Likewise devices `k,l,m...` can process the preview pixelpipe, devices `o,p,q...` the export pixelpipes and finally devices `x,y,z...` the thumbnail pixelpipes. An empty field means that no OpenCL device may serve this type of pixelpipe.

darktable has an internal numbering system, whereby the first available OpenCL device receives the number 0. All further devices are numbered consecutively. This number, together with the device name, is displayed when you start darktable with `darktable -d openc1`. You can specify a device either by number or by name (upper/lower case and whitespace do not matter). If you have more than one device with the same name you need to use the device numbers in order to differentiate them.

A device specifier can be prefixed with an exclamation mark `!`, in which case the device is excluded from processing a given pixelpipe. You can also use an asterisk `*` as a wildcard, representing all devices not previously explicitly mentioned in that group.

Sequence order within a group matters – darktable will read the list from left to right and whenever it tries to allocate an OpenCL device to a pixelpipe it will scan the devices in that order, taking the first free device it finds.

If a pixelpipe process is about to be started and all GPUs in the corresponding group are busy, darktable automatically processes the image on the CPU by default. You can enforce GPU processing by prefixing the list of allowed GPUs with a plus sign `+`. In this case darktable will not use the CPU but rather suspend processing until the next permitted OpenCL device is available.

darktable’s default setting for “openc1\_device\_priority” is `*/!0,*/*/*`.

Any detected OpenCL device is permitted to process the center view image. The first OpenCL device (0) is not permitted to process the preview pixelpipe. As a consequence, if there is only one GPU available on your system, the preview pixelpipe will always be processed on the CPU, keeping your single GPU exclusively for the more demanding center image view. This is a reasonable setting for most systems. No such restrictions apply to export and thumbnail pixelpipes.

The default is a good choice if you have only one device. If you have several devices it forms a reasonable starting point. However, as your devices might have quite different levels of processing power, it makes sense to invest some time optimizing your priority list.

Here is an example. Let’s assume we have a system with two devices, a fast Radeon HD7950 and an older and slower GeForce GTS450. darktable (started with `darktable -d openc1`) will report the following devices:

```
[opengl_init] successfully initialized.
[opengl_init] here are the internal numbers and names of
                OpenCL devices available to darktable:
[opengl_init]      0      'GeForce GTS 450'
[opengl_init]      1      'Tahiti'
[opengl_init] FINALLY: opengl is AVAILABLE on this system.
```

Here, the GeForce GTS 450 is detected as the first device and the Radeon HD7950 ('Tahiti') as the second. This order will normally not change unless the hardware or driver configuration is modified, but it's better to use device names rather than numbers to be on the safe side.

As the GTS450 is slower than the HD7950, an optimized "opengl\_device\_priority" could look like: !GeForce GTS450,\*/!Tahiti,\*/Tahiti,\*/Tahiti,\*.

The GTS450 is explicitly excluded from processing the center image pixelpipe; this is reserved to "all" other devices (i.e. the HD7950/Tahiti). Conversely, for the preview pixelpipe, the Tahiti is excluded, so that only the GTS450 is permitted to do the work.

For file export and thumbnail generation we want all hands on deck. However, darktable should first check whether the Tahiti device is free, because it's faster. If it is not free, then all other devices - in fact only the GTS450 - are checked.

## 12.3.10. opengl still does not run for me

As has been mentioned, OpenCL systems come with a huge variety of setups: different GPU manufacturers and models, varying amounts of GPU memory, different drivers, different distributions etc..

Many of the potential problems will only appear with very specific combinations of these factors. As the darktable developers only have access to a small fraction of these variations, please understand that we might not be able to fix your specific problem. There is not much we can do if we are unable to reproduce your issue.

If nothing else helps, the best option is probably to start darktable with

```
darktable --disable-opengl
```

In the end there is nothing in darktable that only runs on GPU. Don't let the lack of OpenCL discourage you - darktable's CPU code is also highly optimized for performance!

## 12.4. using darktable-chart

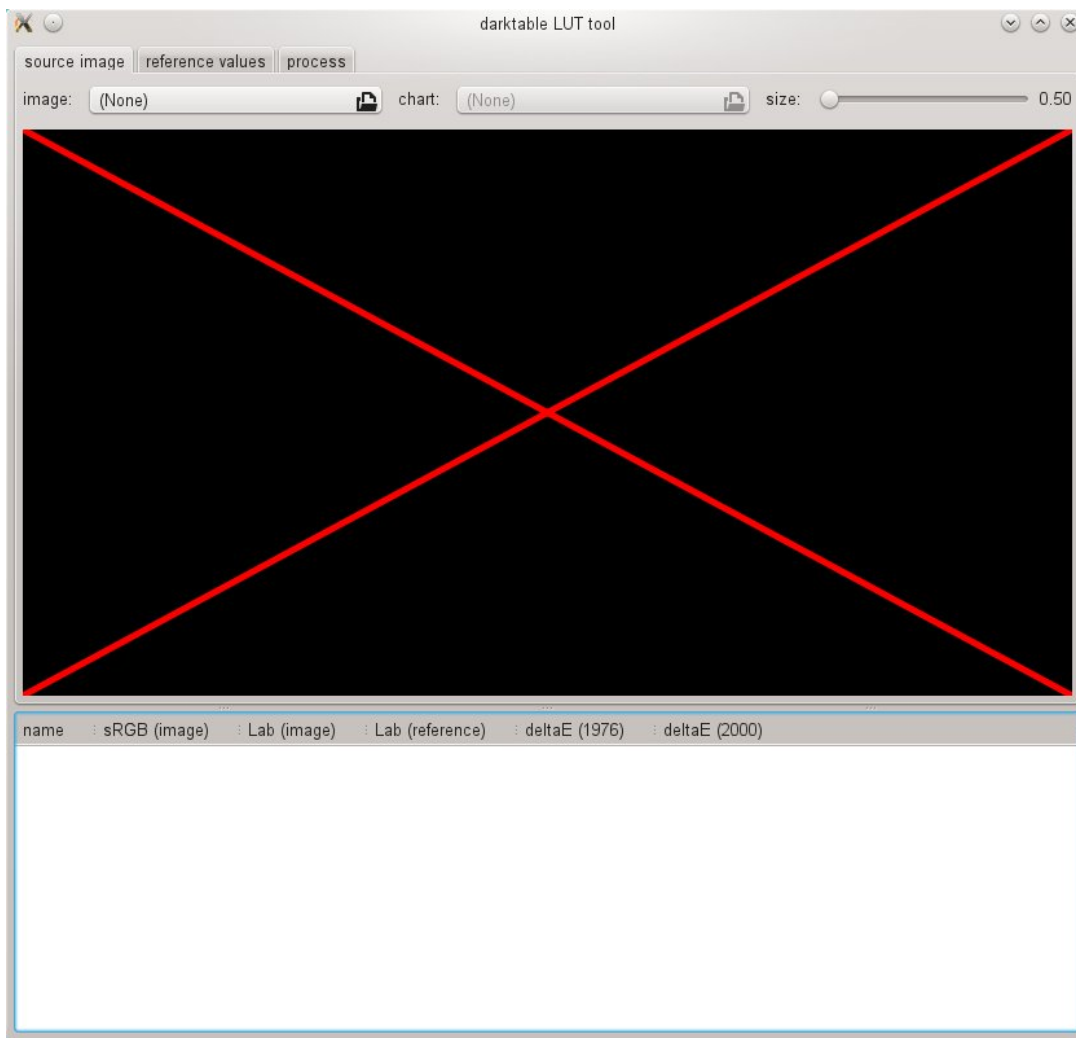
### 12.4.1. overview

darktable-chart is a tool for extracting luminance and color values from images of color reference cards such as IT8.7/1 charts. Its main purpose is to compare a source image (typically a largely unprocessed raw image) to a target image (typically a JPEG image created in-camera) and produce a darktable style that is able to use the luminance and color values of the source image to produce the target image. This style employs the [tone curve](#) module, the [input color profile](#) module, and the [color look up table](#) module for that purpose.

Some cameras offer various film simulation modes of your choice. With the help of darktable-chart and the underlying modules you can create styles that replicate these film simulations from within darktable.

### 12.4.2. usage

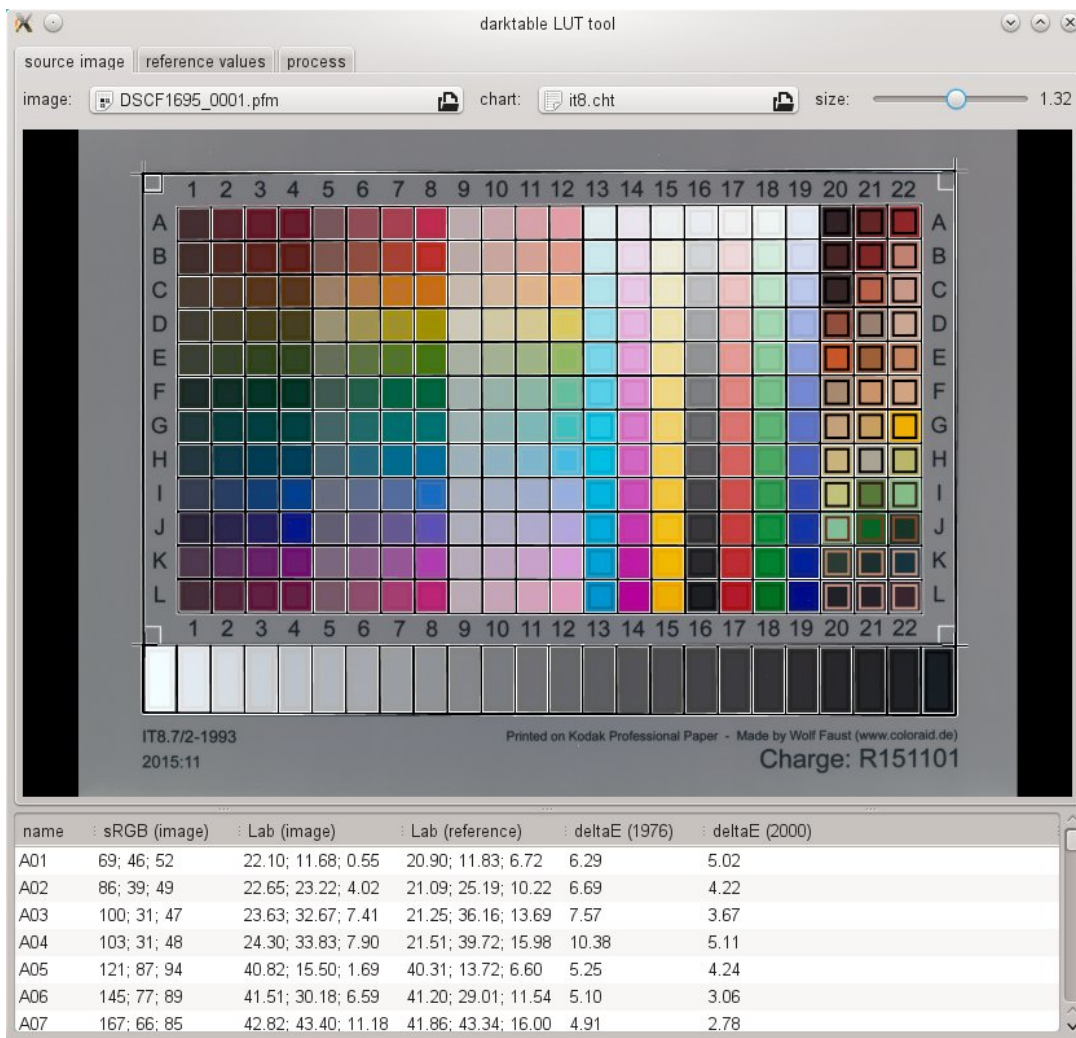
The tool is organized into three tabs in the upper part and a text output frame in the lower part.



The first tab is used to define the source image, the second tab defines the reference (target) and the third tab contains the controls to generate the resulting darktable style.

### 12.4.3. source image

In the “source image” tab you set your source image, which requires two elements. The first element is an input file in Lab Portable Float Map format (extension `.pfm`). The source file represents the largely unmodified data as the camera sees it. Information about how to take photos of a color reference card and produce a `.pfm` output file are described below. The second element is a chart file that contains a formal description of the underlying color reference card’s layout (extension `.cht`). Chart files are usually shipped with your color reference card or can be downloaded from the internet.



In real life the photo taken from the color reference card will show some perspective distortions relative to the layout defined in the chart file. For that reason the layout is displayed as a grid over the image and can be modified.

You can move the corners of the grid using the mouse to reach best alignment of grid and image.

A rectangular frame is displayed for each patch and defines the area from which darktable-chart will sample the required input data. You may need to modify the size of these rectangles so that the sampling area is big enough but does not overlap with neighboring patches. Use the “size” slider in the upper right part of the GUI. Higher values lead to smaller sizes.

### 12.4.4. reference values

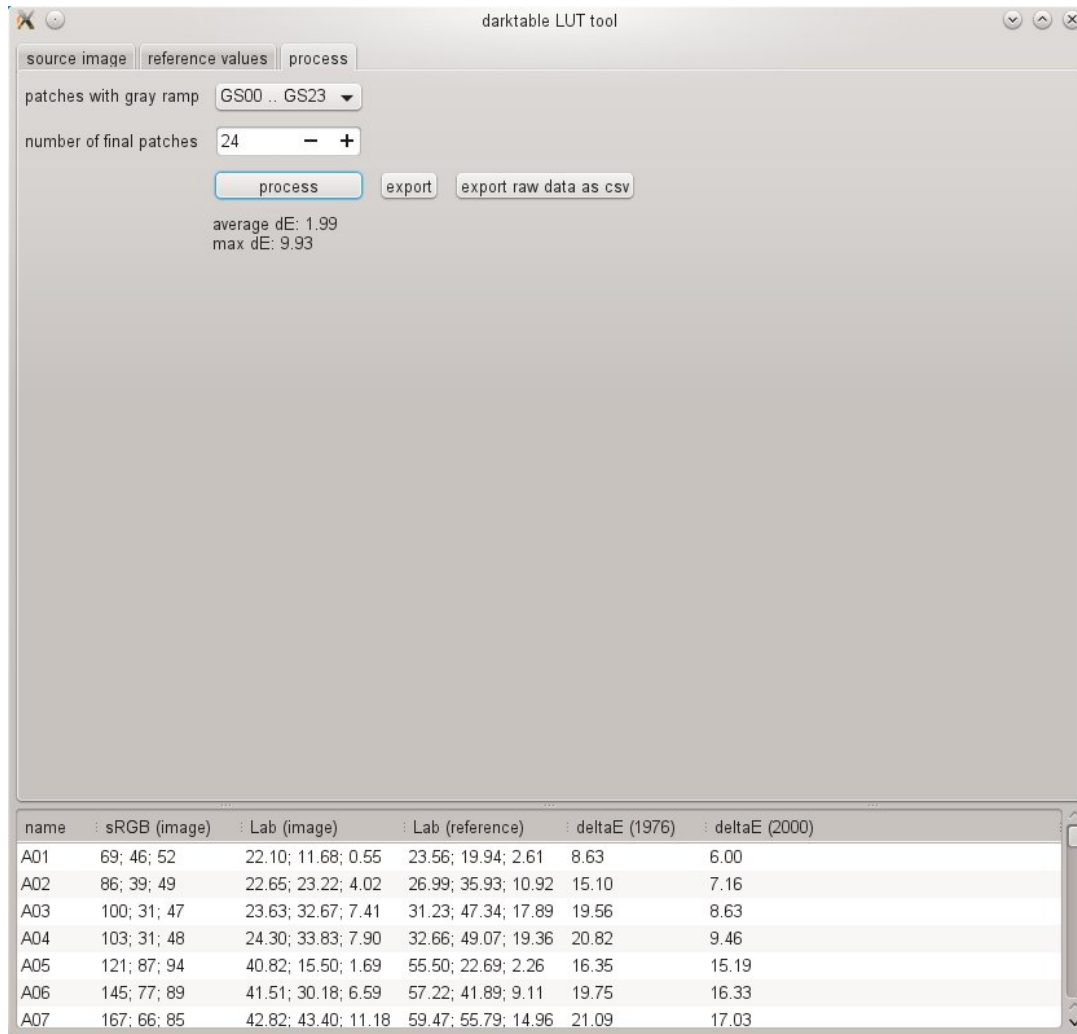
The “reference values” tab determines the target values to which the source image must be modified by the resulting style. You can either supply reference values in the form of measured data of your color reference card (mode “cie/it8 file”), or you can supply a photographic image (mode “color chart image”) much in the same way as described above. This second image must also be supplied in Lab Portable Float Map format. There is no need to supply the chart file again as darktable-chart takes the same one as defined under “source image”. You only need to again align the layout grid and the image and potentially adjust the “size” slider.

In a typical use case the second image will be based on a JPEG file produced in-camera. This way you can create a style to simulate the in-camera processing within darktable.

In the lower text output frame you will see the color values extracted from the available data for each individual color patch. The first column gives the name of the patch, the second and third column show the corresponding color values of the source image in RGB and Lab format, respectively. The fourth column contains the Lab value coming from the reference (or from the chart file if no reference image has been given). Finally, the fifth and sixth columns display how strongly source and reference values deviate in terms of delta-E values.

## 12.4.5. process

If all of the required settings in the “source image” and “reference values” tabs are ready you can move on to the “process” tab.



First, you need to tell darktable-chart which of the patches represents the gray ramp. In the screenshot displayed above, the gray ramp is positioned in the lower part of the color reference chart, denoted as “GS00...GS23”.

The “number of final patches” input defines how many editable color patches the resulting style will use within the [color look up table](#) module.

Click the “process” button to start the calculation.

The quality of the result (in terms of average delta-E and maximum delta-E) is displayed below the button. This data shows how closely the resulting style (when applied to the source image) will be able to match the reference values – the lower the better.

Once you are happy with the result you can click on “export” to save the generated style.

Supply a style name and a style description under which the style will later appear in darktable. darktable-chart saves the style as a .dtstyle file which can be imported into darktable and shared with others. See [styles](#) .

The “export raw data as csv” button allows you to save the extracted raw data as a CSV file for debugging purposes or later use. darktable-chart offers a command line option to produce a style with the desired number of final patches from a supplied CSV file (see [darktable-chart](#) ).

## 12.4.6. making input images for darktable-chart

To start with, you need a suitable photo of your color reference card in RAW+JPEG format. It goes beyond the scope of this manual to explain the details of how to take this photo, but in a nutshell you need to make the shot on a sunny day around midday with the light source (the sun) shining at an angle onto the card. You need to avoid any glare in the image. The neutral white color patch in the gray ramp (G00) should end up at the L value specified in the description of your card. Often this is L=92 and requires you to overexpose the shot by about 1/3 EV. Ideally you will make several shots with slightly different exposures and later select the right one in darktable. Make sure that the chart fills up most of the frame. Use a lens with a “normal” focal length (e.g. 50mm equivalent) and stop down a bit to avoid vignetting.

You then open the raw file in darktable and disable most modules, especially [base curve](#). Select the standard input matrix in the [input color profile module](#) and disable gamut clipping. Select “camera white balance” in the [white balance](#) module.

There is a special situation if your camera automatically applies some lens corrections (namely vignetting correction) to the resulting JPEG file. In this case you need to activate the [lens correction](#) module in darktable so that raw processing matches the JPEG in this respect. However, since darktable’s vignetting correction may not exactly match the in-camera correction, it is better to disable this correction in-camera if possible.

To output your image go to the [export](#) module in the lighttable.

You will need to select “Lab” as the output color profile. This color space is not visible in the combobox by default. You first need to enable it by setting `allow_lab_output` to TRUE in `$HOME/.config/darktable/darktable.rc`. Alternatively, you can start darktable with:

```
darktable --conf allow_lab_output=true
```

Then select “PFM (float)” as the output format and press “export” to generate the source image file.

You can produce the corresponding reference (target) image from the JPEG in a similar way. This time you will need to disable all modules and export with the “Lab” output color profile in “PFM (float)” format.

## 12.5. program invocation

### 12.5.1. darktable

The darktable binary starts darktable with its GUI and full functionality. This is the standard way to use darktable.

darktable can be called with the following command line parameters:

```
darktable [-d {all,cache,camctl,camsupport,control,dev,
             fswatch,input,lighttable,lua,masks,memory,nan,
             openc1,perf,pwstorage,print,sql,ioporder,
             imageio,undo,signal}]
  [<input file>|<image folder>]
  [--version]
  [--disable-openc1]
  [--library <library file>]
  [--datadir <data directory>]
  [--moduledir <module directory>]
  [--tmpdir <tmp directory>]
  [--configdir <user config directory>]
  [--cachedir <user cache directory>]
  [--localedir <locale directory>]
  [--luacmd <lua command>]
  [--noiseprofiles <noiseprofiles json file>]
  [--d-signal <signal>]
  [--d-signal-act <all,raise,connect,disconnect,print-trace>]
  [--enforce-tiling]
  [--conf <key>=<value>]
  [-t <num openmp threads>]
```

All parameters are optional. In most cases darktable should be started without any additional parameters, in which case darktable uses suitable defaults.

- d**  
**{all, cache, camctl, camsupport, control, dev, fswatch, input, lighttable, lua, masks, memory, nan, opencl, perf, pwstorage}**  
 Enable debug output to the terminal. There are several subsystems of darktable and debugging of each of them can be activated separately. You can use this option multiple times if you want debugging output of more than one subsystem (e.g. `darktable -d opencl -d camctl`).
- d-signal <signal>**  
 If `-d signal` or `-d all` is specified, specify the signal to debug using this option. Specify ALL to debug all signals or specify signal using it's full name. Can be used multiple times.
- d-signal-act <all, raise, connect, disconnect, print-trace>**  
 If `-d signal` or `-d all` is specified, specify the signal action to debug using this option.
- enforce-tiling**  
 Force tiling to be used, even on systems with plenty of memory (for easier debugging of tiling issues).
- <input file>|<image folder>**  
 Optionally supply the name of an image file or folder. If a filename is given darktable starts in darkroom view with that file opened. If a folder is given darktable starts in lighttable view with the content of that folder as the current collection.
- version**  
 Print the darktable version number, a copyright notice, some other useful information, and then terminate.
- disable-opencl**  
 Prevent darktable from initializing the OpenCL subsystem. Use this option if darktable crashes at startup due to a defective OpenCL implementation.
- library <library file>**  
 darktable keeps image information in an sqlite database for fast access. The default location of that database file is `$HOME/.config/darktable/library.db`. Use this option to provide an alternative location (e.g. if you want to do some experiments without compromising your original `library.db`). If the database file does not exist, darktable creates it for you. You may also provide `:memory:` as the library file, in which case the database is kept in system memory – all changes are discarded when darktable terminates.
- Whenever darktable starts, it will lock the library to the current user. It does this by writing the current process identifier (PID) into a lock file `<library file>.lock` next to the library specified. If darktable finds an existing lock file for the library, it will terminate immediately.
- datadir <data directory>**  
 Define the directory where darktable finds its runtime data. The default location depends on your installation. Typical locations are `/opt/darktable/share/darktable/` and `/usr/share/darktable/`.
- moduledir <module directory>**  
 darktable has a modular structure and organizes its modules as shared libraries for loading at runtime. This option tells darktable where to look for its shared libraries. The default location depends on your installation. Typical locations are `/opt/darktable/lib64/darktable/` and `/usr/lib64/darktable/`.
- tmpdir <tmp directory>**  
 Define where darktable should store its temporary files. If this option is not supplied darktable uses the system default.
- configdir <config directory>**  
 Define the directory where darktable stores user-specific configuration. The default location is `$HOME/.config/darktable/`.
- cachedir <cache directory>**  
 darktable keeps a cache of image thumbnails for fast image preview and precompiled OpenCL binaries for fast startup. By default the cache is located in `$HOME/.cache/darktable/`. Multiple thumbnail caches may exist in parallel – one for each library file.
- localedir <locale directory>**  
 Define where darktable can find its language-specific text strings. The default location depends on your installation. Typical locations are `/opt/darktable/share/locale/` and `/usr/share/locale/`.
- luacmd <lua command>**  
 A string containing lua commands to execute after lua initialization. These commands will be run after your “luarc” file. If lua is not compiled-in, this option will be accepted but won't do anything.
- noiseprofiles <noiseprofiles json file>**  
 Provide a json file that contains camera-specific noise profiles. The default location depends on your installation. Typical locations are `/opt/darktable/share/darktable/noiseprofile.json` and `/usr/share/darktable/noiseprofile.json`.

**--conf <key>=<value>**

darktable supports a rich set of configuration parameters defined by the user in `$HOME/.config/darktable/darktable.rc`. You may temporarily overwrite individual settings on the command line with this option – these settings will not be stored in `darktable.rc` on exit.

**-t <num openmp threads>**

limit number of openmp threads to use in openmp parallel sections

## 12.5.2. darktable-cli

The `darktable-cli` binary starts the command line interface variant of darktable which allows images to be exported.

This variant does not open any display – it works in pure console mode without launching a GUI. This mode is particularly useful for servers running background jobs.

`darktable-cli` can be called with the following command line parameters:

```
darktable-cli [<input file or folder>]
              [<xmp file>]
              <output file or folder>
              [--width <max width>]
              [--height <max height>]
              [--bpp <bpp>]
              [--hq <0|1|true|false>]
              [--upscale <0|1|true|false>]
              [--style <style name>]
              [--style-overwrite]
              [--apply-custom-presets <0|1|false|true>]
              [--out-ext <extension>]
              [--import <file or dir>]
              [--icc-type <type>]
              [--icc-file <file>]
              [--icc-intent <intent>]
              [--verbose]
              [--help [option]]
              [--core <darktable options>]
```

The user must supply an input filename and an output filename. All other parameters are optional.

**<input file or folder>**

The name of the input file or folder (containing images) to be exported. If you wish to process multiple images or multiple folders use the `--import` option instead.

**<xmp file>**

The optional name of an XMP sidecar file containing the history stack data to be applied during export. If this option is not provided darktable will search for an XMP file that belongs to the given input file(s).

**<output file or folder>**

The name of the output file or destination folder. The export file format is derived from the file extension or from the `--out-ext` option. You can also use a number of [variables](#) in the output filename. For obvious reasons this parameter is mandatory if you use the program on an image folder containing multiple images. If you specify output folder it is recommended that you also specify the file format with `--out-ext`.

**--width <max width>**

Limit the width of the exported image to the given number of pixels.

**--height <max height>**

Limit the height of the exported image to the given number of pixels.

**--bpp <bpp>**

Define the bit depth of the exported image. Permitted values depend on the output file format.

**Note:** This option is not currently functional. If you need to define the bit depth you need to use the following workaround:

```
--core
--conf plugins/imageio/format/<FORMAT>/bpp=<VALUE>
```

where `<FORMAT>` is the name of the selected output format.



**--hq <0|1|true|false>**

Define whether to use high quality resampling during export (see the [export](#) module reference for more details). Defaults to true.

**--upscale <0|1|true|false>**

Define whether allow upscaling during export. Defaults to false.

**--style <style name>**

Specify the name of a style to be applied during export. If a style is specified, the path to the darktable configuration directory must also be specified (i.e. `--core --configdir ~/.config/darktable`). By default no style is applied.

**--style-overwrite**

The specified style overwrites the history stack instead of being appended to it.

**--apply-custom-presets <0|1|false|true>**

Whether to load `data.db` which contains presets and styles. Disabling this option allows you to run multiple instances of `darktable-cli` at the cost of being unable to use the `--style` option. Defaults to true.

**--out-ext <extension>**

Set the output extension to use. If specified takes precedence over `<output file>`. By default this is extracted from `<output file>`. Defaults to `jpg` if `<output folder>` is specified.

**--import <file or dir>**

Specify input file or folder, can be used multiple times. This option cannot be combined with `<input file or folder>`.

**--icc-type <type>**

Specify the ICC profile type, which is the same as specifying the “output profile” in the [output color profile](#) module. Defaults to “image specified”. Use `--help icc-type` to obtain a list of the supported types. See the [output color profile](#) module reference for a more detailed description of the available options.

**--icc-file <file>**

Specify the ICC profile filename. Defaults to an empty filename.

**--icc-intent <intent>**

Specify the rendering intent. Defaults to “image specified”. Use `--help icc-intent` to obtain a list of the supported intents. See [rendering intent](#) for a more detailed description of the available options.

**--verbose**

Enables verbose output.

**--help [option]**

Prints usage and exits. If `option` is specified, additionally prints usage for the given option.

**--core <darktable options>**

All command line parameters following `--core` are passed to the darktable core and handled as standard parameters. See the [darktable binary](#) section for a detailed description.

## 12.5.3. darktable-generate-cache

The `darktable-generate-cache` binary updates darktable’s thumbnail cache. Invoke this program to generate all missing thumbnails in the background when your computer is idle.

`darktable-generate-cache` can be called with the following command line parameters:

```
darktable-generate-cache
    [-h, --help]
    [--version]
    [--min-mip <0-8>] [-m, --max-mip <0-8>]
    [--min-imgid <N>] [--max-imgid <N>]
    [--core <darktable options>]
```

All parameters are optional. If started without parameters `darktable-generate-cache` uses reasonable defaults.

**-h, --help**

Display usage information and terminate.

**--version**

Display copyright and version information and terminate.

**--min-mip <0-8>, -m, --max-mip <0-8>**

darktable can store thumbnails with up to eight different resolution steps for each image. These parameters define the maximum resolution to be generated (defaults to a range of 0-2). There is normally no need to generate all possible resolutions here – missing ones will be automatically generated by darktable the moment they are needed. When asked to generate multiple resolutions at once, the lower-resolution images are quickly downsampled from the highest-resolution image.

**--min-imgid <N>, --max-imgid <N>**

Specifies the range of internal image IDs from the database to work on. If no range is given, darktable-generate-cache will process all images.

**--core <darktable options>**

All command line parameters following --core are passed to the darktable core and handled as standard parameters. See the [darktable binary](#) section for a detailed description.

## 12.5.4. darktable-chart

The darktable-chart binary is a dedicated utility to create styles out of pairs of images such as RAW+JPEG with in-camera processing. Details about its usage can be found in the [using darktable-chart](#) section.

darktable-chart can either start a GUI or be used as a command-line program.

```
darktable-chart
    [--help]
    [<input Lab pfm file>]
    [<cht file>]
    [<reference cgats/it8 or Lab pfm file>]
```

All parameters are optional, however, if you want to supply the second file name you also need to supply the first one. Starting darktable-chart this way opens a special GUI (details can be found in the [using darktable-chart](#) section).

**--help**

Provide usage information and terminate.

**<input Lab pfm file>**

Open the utility with the given file as source image. The input file needs to be in Lab Portable Float Map format.

**<cht file>**

Specify a chart file describing the layout of the used color reference chart.

**<reference cgats/it8 or Lab pfm file>**

Specify the reference values, either as measured values according to the CGATS standard, or as a reference image in Lab Portable Float Map format.

Alternatively darktable-chart can be used as a command line program to generate darktable style files using previously saved CSV files.

```
darktable-chart
    --csv
    <csv file>
    <number patches>
    <output dtstyle file>
```

All parameters are mandatory.

**<csv file>**

A CSV file previously saved from within darktable-chart.

**<number of patches>**

The number of color patches to be used in the color look up table settings of the created style.

**<output dtstyle file>**

The name of the style file to be created.

## 12.5.5. darktable-cltest

The `darktable-cltest` binary checks if there is a usable OpenCL environment on your system that darktable can use. It emits some debug output that is equivalent to calling `darktable -d opengl` and then terminates.

`darktable-cltest` is called without command line parameters.

## 12.5.6. darktable-cmstest

The `darktable-cmstest` binary (Linux only) investigates whether the color management subsystem of your computer is correctly configured and displays some useful information about the installed monitor profile(s).

`darktable-cmstest` is called without command line parameters.

## 12.5.7. purge\_non\_existing\_images.sh

Find and remove entries from the library database referencing images that no longer exist in the filesystem. You must close darktable before running this script.

The script can be called with the following command line parameters:

```
purge_non_existing_images.sh [-c|--configdir <path>]
                             [-l|--library <path>]
                             [-p|--purge]
```

Run the script with no options to perform a “dry run”, which generates a report of the missing files without committing any changes to the database.

The available options are:

### **-c|--configdir <path>**

Specify the path to the darktable configuration directory to be used by the script. If this option is not provided, the default config directory location will be used.

### **-l|--library <path>**

Specify the path to the `library.db` database file to be analysed by the script. If this option is not specified, the default `library.db` file location will be used.

### **-p|--purge**

Actually delete any entries in the database that refer to non-existent files. If the option is not provided, a report will be printed without committing any changes to the database.

---

### Notes:

1. The script must be run in a unix shell, and the `sqlite3` client must be available in the command search path. For Linux systems, this will normally not be an issue.
  2. For Windows systems, you will normally need the MSYS2 environment to be installed, as described in the [instructions for building darktable in a Windows environment](#). If you installed darktable using the standard Windows installer package, the location of the script would normally be something like: `C:\Program Files\darktable\share\darktable\tools\purge_non_existing_images.sh`.
  3. For macOS systems, the Terminal application provides a shell, and the `sqlite3` client is provided by the operating system by default. If darktable was installed using an application bundle from a dmg image, then the default location for the script would be `/Applications/darktable.app/Contents/Resources/share/darktable/tools/purge_non_existing_images.sh`
  4. The delete operation can't be undone. It is therefore strongly recommended that you take a backup of the database before purging any entries.
-

## 12.6. variables

darktable supports variable substitution in a number of modules and preference settings. For example:

- Defining file names in the [export](#) module
- Displaying image information in the darkroom's [image information line](#)
- Displaying image information in the lighttable's overlays and tooltips (see [preferences > lighttable](#) )
- Placing text on an image in the [watermark](#) processing module

## available variables

The following variables are available, though they may not all be applicable in every context:

<code>\$(ROLL_NAME)</code>	film roll of the input image
<code>\$(FILE_FOLDER)</code>	folder containing the input image
<code>\$(FILE_NAME)</code>	basename of the input image
<code>\$(FILE_EXTENSION)</code>	extension of the input image
<code>\$(ID)</code>	the image id
<code>\$(VERSION)</code>	the duplicate version number
<code>\$(VERSION_IF_MULTI)</code>	same as <code>\$(VERSION)</code> but null string if only one version exists
<code>\$(VERSION_NAME)</code>	version name from metadata
<code>\$(SEQUENCE)</code>	a sequence number within an export job
<code>\$(SENSOR_WIDTH)</code>	width of RAW data in pixels before RAW crop
<code>\$(SENSOR_HEIGHT)</code>	height of RAW data in pixels before RAW crop
<code>\$(RAW_WIDTH)</code>	width of RAW data in pixels after RAW crop
<code>\$(RAW_HEIGHT)</code>	height of RAW data in pixels after RAW crop
<code>\$(CROP_WIDTH)</code>	image width in pixels at the end of the pixelpipe, but before export resize
<code>\$(CROP_HEIGHT)</code>	image height in pixels at the end of the pixelpipe, but before export resize
<code>\$(EXPORT_WIDTH)</code>	image width in pixels at the end of the pixelpipe and after export resize
<code>\$(EXPORT_HEIGHT)</code>	image height in pixels at the end of the pixelpipe and after export resize
<code>\$(MAX_WIDTH)</code>	maximum width entered in export module
<code>\$(MAX_HEIGHT)</code>	maximum height entered in export module
<code>\$(YEAR)</code>	year at date of export
<code>\$(MONTH)</code>	month at date of export
<code>\$(DAY)</code>	day at date of export
<code>\$(HOUR)</code>	hour at time of export
<code>\$(MINUTE)</code>	minute at time of export
<code>\$(SECOND)</code>	second at time of export
<code>\$(EXIF_YEAR)</code>	Exif year
<code>\$(EXIF_MONTH)</code>	Exif month
<code>\$(EXIF_DAY)</code>	Exif day
<code>\$(EXIF_HOUR)</code>	Exif hour
<code>\$(EXIF_MINUTE)</code>	Exif minute
<code>\$(EXIF_SECOND)</code>	Exif second
<code>\$(EXIF_ISO)</code>	ISO value
<code>\$(EXIF_EXPOSURE)</code>	Exif exposure
<code>\$(EXIF_EXPOSURE_BIAS)</code>	Exif exposure bias
<code>\$(EXIF_APERTURE)</code>	Exif aperture
<code>\$(EXIF_FOCAL_LENGTH)</code>	Exif focal length
<code>\$(EXIF_FOCUS_DISTANCE)</code>	Exif focus distance
<code>\$(LONGITUDE)</code>	longitude
<code>\$(LATITUDE)</code>	latitude
<code>\$(ELEVATION)</code>	elevation
<code>\$(STARS)</code>	star rating (text only)
<code>\$(RATING_ICONS)</code>	star rating (using star characters)
<code>\$(LABELS)</code>	colorlabels (text only)
<code>\$(LABELS_ICONS)</code>	colorlabels (using colored bullet characters)
<code>\$(LABELS_COLORICONS)</code>	colorlabels (using colored icons)
<code>\$(MAKER)</code>	camera maker
<code>\$(MODEL)</code>	camera model
<code>\$(LENS)</code>	lens
<code>\$(TITLE)</code>	title from metadata

---

<code>\$(DESCRIPTION)</code>	description from metadata
<code>\$(CREATOR)</code>	creator from metadata
<code>\$(PUBLISHER)</code>	publisher from metadata
<code>\$(RIGHTS)</code>	rights from metadata
<code>\$(TAGS)</code>	tags list (Xmp.dc.Subject)
<code>\$(CATEGORYn(category))</code>	tag name of level n [0,9] of selected category (or tag)
<code>\$(SIDECAR_TXT)</code>	content of the text sidecar file (if any)
<code>\$(PICTURES_FOLDER)</code>	pictures folder
<code>\$(HOME)</code>	home folder
<code>\$(DESKTOP)</code>	desktop folder
<code>\$(OPENCL_ACTIVATED)</code>	whether OpenCL is activated
<code>\$(USERNAME)</code>	user name defined by OS
<code>\$(NL)</code>	newline character
<code>\$(JOBCODE)</code>	internal jobcode of current job

## string substitution

All of the variables support basic string substitution inspired by bash though some of the details differ.

All patterns are treated as simple string comparisons. There is no regex support.

The following string replacement functions are provided, where var is one of the variables listed above:

<code>\$(var-default)</code>	If var is empty, return "default"
<code>\$(var+alt_value)</code>	If var is set, return "alt_value" else return empty string
<code>\$(var:offset)</code>	Return var starting from offset If offset is negative count from the end of the string
<code>\$(var:offset:length)</code>	Starting from offset, return at most length characters of var If offset is negative the length is counted from the end of var If length is negative this indicates the end of the result, counted from the end of var, and not an actual length
<code>\$(var#pattern)</code>	Remove "pattern" from the start of var
<code>\$(var%pattern)</code>	Remove "pattern" from the end of var
<code>\$(var/pattern/replacement)</code>	Replace the first occurrence of "pattern" in var with "replacement" If "replacement" is empty then "pattern" will be removed
<code>\$(var//pattern/replacement)</code>	Replace all occurrences of "pattern" in var with "replacement" If "replacement" is empty then "pattern" will be removed
<code>\$(var/#pattern/replacement)</code>	If var starts with "pattern" then "pattern" is replaced with "replacement"
<code>\$(var/%pattern/replacement)</code>	If var ends with "pattern" then "pattern" is replaced with "replacement"
<code>\$(var^)</code>	Make the first character of var uppercase
<code>\$(var^^)</code>	Make all characters of var uppercase
<code>\$(var,)</code>	Make the first character of var lowercase
<code>\$(var,,)</code>	Make all characters of var lowercase

## 12.7. default module order

The following sections describe the default module order in the new scene-referred workflow and the legacy display-referred workflow. Note that in the following sections the module order goes from top (input file) to bottom (output image).

## scene-referred module order

The default ordering of modules when using the new scene-referred workflow is as follows:

1. **raw black/white point**
2. [invert \(deprecated\)](#)
3. **white balance**
4. **highlight reconstruction**
5. [raw chromatic aberrations](#)
6. [hot pixels](#)
7. [raw denoise](#)
8. [demosaic](#)
9. [denoise \(profiled\)](#)
10. [surface blur](#)
11. [rotate pixels](#)
12. [scale pixels](#)
13. [lens correction](#)
14. [chromatic aberrations](#)
15. [haze removal](#)
16. [rotate and perspective](#)
17. **orientation**
18. [crop and rotate \(deprecated\)](#)
19. [liquify](#)
20. [spot removal \(deprecated\)](#)
21. [retouch](#)
22. **exposure**
23. [local tone mapping \(deprecated\)](#)
24. [tone equalizer](#)
25. [crop](#)
26. [graduated density](#)
27. [unbreak input profile](#)
28. [legacy equalizer](#)
29. **input color profile**
30. [color calibration](#)
31. [diffuse or sharpen](#)
32. [censorize](#)
33. [negadoctor](#)
34. [astrophoto denoise](#)
35. [color look up table](#)
36. [defringe \(deprecated\)](#)
37. [contrast equalizer](#)
38. [lowpass](#)
39. [highpass](#)
40. [sharpen](#)
41. [color transfer \(deprecated\)](#)
42. [color mapping](#)
43. [channel mixer \(deprecated\)](#)
44. [basic adjustments \(deprecated\)](#)
45. [color balance](#)
46. [color balance rgb](#)
47. [rgb curve](#)
48. [rgb levels](#)
49. [base curve](#)
50. [filmic \(legacy\)](#)
51. **FILMIC RGB** - *transition from scene-referred to display-referred space*
52. [lut 3D](#)
53. [contrast brightness saturation](#)
54. [tone curve](#)
55. [levels](#)
56. [shadows and highlights](#)
57. [zone system \(deprecated\)](#)
58. [global tonemap \(deprecated\)](#)
59. [fill light \(deprecated\)](#)
60. [local contrast](#)
61. [color correction](#)
62. [color contrast](#)
63. [velvia](#)
64. [vibrance \(deprecated\)](#)

- 
65. [color zones](#)
  66. [bloom](#)
  67. [colorize](#)
  68. [lowligh vision](#)
  69. [monochrome](#)
  70. [grain](#)
  71. [soften](#)
  72. [split-toning](#)
  73. [vignetting](#)
  74. [color reconstruction](#)
  75. **[output color profile](#)**
  76. [dithering](#)
  77. [framing](#)
  78. [watermark](#)

**key:**

- *italic*: not recommended for scene-referred workflow
- **bold**: module on by default



## legacy/display-referred module order

The default ordering of modules when using the legacy display-referred workflow is as follows:

1. **raw black/white point**
2. [invert \(deprecated\)](#)
3. **white balance**
4. **highlight reconstruction**
5. [raw chromatic aberrations](#)
6. [hot pixels](#)
7. [raw denoise](#)
8. [demosaic](#)
9. [denoise \(profiled\)](#)
10. [local tone mapping \(deprecated\)](#)
11. [exposure](#)
12. [spot removal \(deprecated\)](#)
13. [retouch](#)
14. [lens correction](#)
15. [chromatic aberrations](#)
16. [rotate and perspective](#)
17. [liquify](#)
18. [rotate pixels](#)
19. [scale pixels](#)
20. [orientation](#)
21. [crop and rotate \(deprecated\)](#)
22. [tone equalizer](#)
23. [crop](#)
24. [graduated density](#)
25. **BASE CURVE** – *default transition between scene-referred and display-referred space*
26. [surface blur](#)
27. [unbreak input profile](#)
28. [haze removal](#)
29. **input color profile**
30. [color calibration](#)
31. [diffuse or sharpen](#)
32. [censorize](#)
33. [negadoctor](#)
34. [basic adjustments \(deprecated\)](#)
35. [color reconstruction](#)
36. [color look up table](#)
37. [defringe \(deprecated\)](#)
38. [legacy equalizer \(deprecated\)](#)
39. [vibrance \(deprecated\)](#)
40. [color balance](#)
41. [color balance rgb](#)
42. [colorize](#)
43. [color transfer \(deprecated\)](#)
44. [color mapping](#)
45. [bloom](#)
46. [astrophoto denoise](#)
47. [global tonemap \(deprecated\)](#)
48. [shadows and highlights](#)
49. [contrast equalizer](#)
50. [local contrast](#)
51. [color zones](#)
52. [lowligh vision](#)
53. [monochrome](#)
54. [filmic \(legacy\)](#)
55. [filmic rgb](#)
56. [lut 3D](#)
57. [contrast brightness saturation](#)
58. [zone system \(deprecated\)](#)
59. [tone curve](#)
60. [levels](#)
61. [rgb levels](#)
62. [rgb curve](#)
63. [fill light \(deprecated\)](#)
64. [color correction](#)

- 
65. [sharpen](#)
  66. [lowpass](#)
  67. [highpass](#)
  68. [grain](#)
  69. [color contrast](#)
  70. [output color profile](#)
  71. [channel mixer \(deprecated\)](#)
  72. [soften](#)
  73. [vignetting](#)
  74. [split-toning](#)
  75. [velvia](#)
  76. [dithering](#)
  77. [framing](#)
  78. [watermark](#)

## 12.8. darktable's color pipeline

Most image processing applications come from the 1990s and/or inherit a 1990s workflow. These applications processed images encoded with 8 bit unsigned integers because it was more memory- and computationally-efficient. However, due to the use of an integer format (which implies rounding errors) they had to apply a “gamma” (essentially a transfer function applying a power  $1/2.2$  or  $1/2.4$  to encode the RGB values) and increase the bit-depth in the low-lights in order to reduce rounding errors there (humans are very sensitive to low-light details). The 8 bit integer formats are also technically limited to the 0-255 range. Anything outside of this range overflows and is clipped to the nearest bound.

These workflows, using bounded RGB representations and possibly non-linear transforms to encode RGB signals, are called “display-referred”. They rely on the assumption that the image has been prepared for display at an early stage in the processing pipeline, and embed hard-coded assumptions about the RGB values of black, middle-gray and white. Most of the image-processing algorithms used in these workflows have been tuned around these assumptions. For example, the overlay blending mode expects a middle-gray encoded at 50% (or 128 in integer encoding).

Unfortunately the non-linear scaling, which is mandatory to make the integer encoding work, breaks the natural relationships between pixel values. Hue and saturation change in unpredictable ways, and value relationships between neighbouring pixels are dilated or compressed such that gradients are also altered unpredictably.

Display-referred pipelines therefore break optical filters (lens blurring or deblurring), alpha compositing (which relies on optical and geometrical definitions of occlusion), colors and gradients (local relationships between chrominance and luminance of pixels). They also don't scale well to HDR images, which led to the development of many questionable local and global tonemapping methods and the infamous 2010s “HDR look”.

Modern computers are not tied to the same computational limitations as those from the 1990s, and can work on pixels whose values are completely unbounded (from 0 up to +infinity) and encoded as real numbers (using floating point formats). These possibilities enable what we call a “scene-referred” workflow, in which pixels can retain their original radiometric relationships along almost the entire processing pipe. In scene-referred workflow, pixels are prepared for display only at the last stage of the pipeline, in the display transform. This means that the RGB values of the pixels are kept proportional to the intensity of the light emission recorded by the camera on the scene, enabling accurate alpha compositing and optical filter emulations, while also scaling to any dynamic range through the same algorithm (SDR as well as HDR).

However, scene-referred pipelines lose the convenient fixed values of white, middle-gray and black which characterised display-referred pipelines, and setting these values, according to the scene and to the conditions of shooting, now becomes the responsibility of the user. This requires a more complex user interface.

Also, because scene-referred values are supposed to be physically-meaningful, pixels cannot have zero intensity. This would mean they have no light at all, and the existence of zero light breaks many physically-accurate algorithms. In fact, white and black mean nothing with regard to the original scene, which is only a collection of luminances at varying intensities. The scene-referred workflow simply aims at remapping some arbitrary scene luminances to what will appear white or black on the output medium.

Versions of darktable prior to 2.6 had a non-linear display-referred pipeline, assuming that a non-linear transform took place early in the pipe and middle-gray was thereafter encoded as 50%. However, not all modules and filters clipped pixel values above 100%, leaving open the possibility of recovering those values later in the pipe.

The *filmic* module's view transform, introduced in darktable 2.6, was the first step toward a scene-referred pipeline, and deferred the mandatory, non-linear, display preparation to the end of the pipe, along with the ability to set custom black, gray and white values. The *color balance* module then introduced a way to deal with a variable definition of middle-gray.

Starting in darktable 3.2, users could choose between two workflows that defined consistent default settings, modules and pipeline order for both *display-referred* and *scene-referred* processing.

In darktable 3.4, a full scene-referred masking and blending option was introduced, allowing masks to be defined for pixel values above 100% and using only unbounded blending operators.

Switching to *scene-referred* is a cognitive leap for most experienced users, who are used thinking in display-referred ways. In a display-referred workflow, it is customary to anchor the white value and let tone adjustments revolve around that point, trying to maximize brightness while avoiding clipping. In a scene-referred workflow, white and black values are fluid and adapted to the output medium. It is advised that users anchor middle-gray (which will be preserved as-is for any output medium) and let the view transform (*filmic*) dilate or contract the dynamic range around that point. Because 10 bit HDR white is 4 times as bright as 8 bit SDR white, any rigid definition of “white” becomes irrelevant. But anchoring for middle-gray is actually more convenient, since it keeps the average brightness of the picture unchanged through the view transform.

Some modules (*levels*, *rgb levels*, *tone curve*, *rgb curve*) are inherently incompatible with a scene-referred workflow, because their graphical interface implicitly suggests RGB values that are bounded within the 0-100% range. While the pixel operations they perform can be used in either scene-referred or display-referred workflows because they are unbounded internally, their controlling interface does not allow pixels to be selected outside of the 0-100% range.

Similarly, blending modes such as overlay, linear light, soft light, hard light, darken, brighten, etc. all have hard-coded thresholds that internally expect display-referred non-linear encoding.

In darktable 3.4 and above, hovering the cursor over a module header shows a tooltip detailing the color spaces, ranges and encodings that the module expects, uses and produces. Here are the definitions of the terms used:

#### **linear**

Pixel values are proportional to the scene radiometric emission, in a way that enables accurate emulation of physical filters.

#### **non-linear**

Pixel values are re-scaled such that low-lights are given a larger encoding range, usually to remap the 18.45% reference middle-gray to a value between 46 and 50%.

#### **display-referred**

Pixel values are expected to lie between 0 and 100% of the display range, where 100% is understood to be the luminance of a 20% reflective white surface (the white patch of a Color Checker) and 0% is understood to be the maximum density of the output medium (saturated black ink or minimum LED panel backlighting).

#### **scene-referred**

Pixel values are expected to be greater than zero up to +infinity. The meaning of specific pixel values needs to be defined at runtime by the user. Scene-referred values don't automatically imply a linear, radiometrically scaled, encoding.

## 12.9. midi device support

The shortcut mapping system has been most extensively tested with the Behringer X-Touch Mini and contains custom code to deal with the specific features of this device. All other devices are treated as “generic midi” and may or may not work (well) or need some manual setup each time. A configuration framework (possibly in lua) to support additional device-specific features is on the to do list.

### Behringer X-touch Mini

The device should be in Standard Mode (not MC). Layers A & B are somewhat supported, however, since the device does not send a notification when switching between layers, and since the updates to “hidden” rotors and lights (both under buttons and the pattern used around the rotors) are ignored, everything will only be updated completely after you press or turn something in the “new” layer.

The light rings around the rotors show a different pattern depending on whether the associated slider has a symmetrical +/- range centered around 0, 0-100 (or 0-1) or anything else. If linked to a drop-down (or star rating) the rotor will step one light at the time. If the drop-down has many (more than 13) items (like the blending mode drop-down), the rotor will go around twice; the first time with one light on, and the second time with two adjacent lights.

### Arturia Beatstep

Individual rotors can be configured to send absolute (0-127) values or changes (+/- 1,2,3,... in different encodings). The recommended setting is Relative #1 for all knobs with Knob Acceleration set to Slow (Off) or Medium. This can be configured with [Midi Control Center](#), available for Windows or MacOS. By default the darktable midi module assumes, for unknown devices, that rotors use absolute encodings. After startup it therefore needs to be told that the device is sending relative movements. You can do this by turning one of the encoders slowly left (down) 5 times. This allows the module to determine which of the several relative encodings is used. If you don't succeed switching encodings on the first attempt, you can reinitialise the input modules by pressing Ctrl+Shift+Alt+I and try again. If you still have problems, you could try [this](#).

If you map the Beatstep's buttons to ratings, color labels or toggle buttons, they should light up when the corresponding element is switched on.

## Behringer BCR2000

(and possibly BCF2000)

These machines are highly configurable so there are many settings that could complicate the interaction with darktable's midi module. The [BC Manager tool](#) (available for Windows and MacOS) can be used to configure them. The easiest thing to do is to reset all encoders and buttons to their simplest settings, which can be done (for the BCR2000) using [this file](#). You can send it to the machine with BC Manager or (under Linux) with amidi. There's also a global setting called "Deadtime" that determines how long the BCR ignores arriving messages after sending out updates. This is to avoid feedback loops, but for darktable it means that it blocks the adjustments sent back immediately after each rotor move. So Deadtime needs to be set to 0.

Once configured in this way, button and rotor lights should all work. The top row of rotors (or the only row for the BCF2000) support different light patterns (like the X-Touch Midi, with some extra options and it has 15 instead of 13 leds). However, configuring these is more complicated, and not yet implemented. It also seems to be slower, so may not be doable in real-time for each update (like it is done for the X-touch). So at the moment only the 12-dot pattern is used. For rotors linked to drop-down lists, just like on the X-touch the rotor, the light goes round twice; for the first 15 options only one light goes on, for the second 15 options, two adjacent lights will be on.

## Loupedeck

(and Loupedeck+ but not the later non-midi devices)

Going by the information provided [here](#), most of the functionality of the first two loupedeck models (both midi based) should be supported, with the following exceptions and caveats:

- the Hue/Sat/Lum buttons can be mapped to the notebook tabs in the color zones module but the little lights next to them may not respond to update messages sent when the user switches tabs using the mouse, so they might get out of sync
- the 8 sliders/wheels can be individually mapped to the elements of the graph in color zones

Other than that, since the loupedeck uses Relative encoding (so is similar to the Beatstep) it will need to tell the midi module at each startup (until a configuration system has been implemented). You do this by turning any of the knobs (or wheels) slowly down/left 5 times. Re-initialise the input layer with Ctrl+Shift+Alt+I if it didn't work the first time so you can try again.

## Korg nanoKONTROL2

The device should be configured first using the Korg Kontrol Editor application to be in the CC mode and every button should be set to the Note type and Momentary button behavior. To control the lights in the buttons the LED mode should be set to External. It is important to note that the Track and the Marker buttons do not have leds in them.

There is an Kontrol Editor profile available [here](#) which can be loaded using the Windows application to directly configure all these settings to correctly work with darktable.

## 12.10. contributing to dtdocs

This page defines the style guide for dtdocs and information about how to contribute to the project.

It is included in the user manual so that you can see how the page is rendered as well as how it is written. Please go to [GitHub](#) to see the source for this page.

The manual structure and content have been carefully considered based on the following criteria:

1. The manual should be comprehensive – it should describe all of the functionality available in darktable
2. It should have a consistent and logical structure and every piece of functionality should have its own logical place within that structure
3. It should be as long as necessary but as short as possible – brevity is a must
4. It should be objective
5. Functionality should be explained once and only once (with the exception of the basic workflow guidelines in the overview section)
6. Images should be included only where necessary to improve understanding of key principles and should not contain text unless it is unavoidable

We are generally **not** interested in:

1. Restructuring the manual
2. Switching markup languages
3. Detailed workflow tutorials (though we are interested in publishing those on the blogs of either darktable.org or pixls.us)

We **are** interested in

1. Spelling and grammar corrections
2. Clarification of text
3. Documentation for new features

We are always extremely interested in hearing about which sections of the manual did not make sense to you and *why*, so that we can improve the documentation.

In general, if you wish to make a major change, please open an issue and discuss it with the maintainers first. This is to avoid doing work that wouldn't be accepted.

## format

This website is authored in pure markdown, using some extensions. It is initially designed to work with the Hugo SSG but intended to be portable enough that it can be easily rendered with another application if required.

Files should be rendered in UTF-8 and should not include any column wrapping.

## structure

The following shows the structure of an example main chapter with subsections in the dtdocs website.

```
example-chapter/  
  _index.md  
  section1-with-subsections/  
    subsection1/  
      image.png
```



```

_index.md
subsection1.md
subsection2.md
section2.md
section3.md

```

A couple of notes on the above structure:

- `_index.md` files do not contain any content (they contain metadata only) and are used to render section headers and ToC entries. In the above example `example-chapter/_index.md` defines the title of the example chapter and the order in which it appears in the main table of contents. Similarly `example-chapter/section1-with-subsections/_index.md` defines metadata for the first section of the chapter.
- Media files should be contained in a directory with the same name as the page to which they relate. In this example, `example-chapter/section1-with-subsections/subsection1` contains media related to the `subsection1.md` page.

## metadata

Metadata for the markdown files is presented at the head of the page using yaml. Any metadata may be defined – the module reference sections contain quite a lot of specific metadata – however the following defines some minimal metadata for the example page `example-chapter/section1-with-subsections/subsection1.md`.

```

---
title: Sub Section 1 Title
id: subsection1
weight: 10
---

```

### title

This should contain the rendered title of your page. To include a colon within a title, enclose the title in double-quotes.

### id

This is the id used to identify the page by Hugo. It should usually be the same name as the file (for content files) or the parent directory (for `_index.md` files).

### weight

This is an optional metadata field used to define the order in which sections are presented in the Table of Contents. If the *weight* field is not included, pages will be rendered in alphabetic order by default. For example, to define the sections and subsections of the above example in reverse order, the following metadata would need to be set:

```

example-chapter/
  section1-with-subsections/
    _index.md           # weight: 30 (place section1 page at the end of example-chapter)
    subsection1.md     # weight: 20 (place subsection1 page at the end of section1)
    subsection2.md     # weight: 10 (place subsection2 page at the start of section1)
  section2.md         # weight: 20 (place section2 in the middle of example-chapter)
  section3.md         # weight: 10 (place section3 at the start of example-chapter)

```

# content

## general style guidance

- All content should be authored in plain markdown without shortcodes and HTML should be kept to an absolute minimum , if used at all
- Minimalism is an absolute must. Fewer words are preferred
- Markdown files should be as short as possible
- Follow the naming and capitalization norms present in the GUI of the application – namely all headers and titles are in lower case, except for the very top-level chapter names
- Headers in a file should not exceed level three (###)
- The primary authoring language is English
- Assume the reader has the application open while reading the user manual and only include images where they contribute to the explanation of complex functionality
- Use image callouts if you need to annotate an image (i.e. mark parts of the image with a letter or number and then explain the meaning in some text following the image). Do not place words directly into the image for annotations, as this makes localization difficult. See [this page](#) for an example.
- Changes to the content should be proposed via pull request or a similar mechanism
- Your submissions will be copy-edited – don't take it personally

## definition lists

The standard method of presenting information about darktable module controls is with the use of definition lists.

### gui control name

A declaration of what the control does. For example “Set the exposure in EV units”.

You can include as many paragraphs as you like, but try to limit to 2 or 3 where possible.



### a control accessed through a button with an icon

When a control is activated using an icon, take a screenshot of the icon using the standard darktable theme and add it before the name of the control

### gui combobox name

Comboboxes often have multiple options that all need to be displayed with separate definitions. Use bulleted lists with *italics* for the combobox values.

- *the first value*: What the first value means
- *the second value*: What the second value means

Definition lists are also used throughout the document, wherever a named piece of functionality needs to be defined. See, for example, [darktable-cli](#) .

## notes

If you wish to present an important note to the user, use the following format:

---

**Note:** This is an important note.

---

## fixed-width fonts and code blocks

Fixed width fonts (using the ``` character) should normally only be used for code blocks and when referencing file names and command line parameters.

## links

Internal links must be relative to the current file and must point to a valid markdown (.md) file. Start links with either `./` to represent the current directory or `../` to represent the parent directory.


- Links to a processing module should be in italics, e.g. [\*exposure\*](#)
- Links to a utility module should be in plaintext, e.g. [history stack](#)
- Link to a top level section by referencing the `_index.md` file, e.g. [module reference](#)
- Link to a tab in the preferences dialog: [preferences > general](#)
- Link to a specific preference setting: [preferences > general > interface language](#)
- Each header within a page can be linked to directly with an anchor link: [contributing/notes](#)

## images

When taking screenshots from the darktable application itself, use the default darktable theme.

Several filename suffixes can be used to control how an image is rendered.

### icon

To insert an image as an icon, include `#icon` after the image name in the link. The markdown `![squirrel icon](./contributing/contributing.png#icon)` outputs the following: 

### image width

You can set the image width to 25, 33, 50, 66, 75 or 100 per cent of the rendered page width by including `#wxx` after the image name in the link, where `xx` is the desired width. For example:

`![squirrel](./contributing/squirrel.png#w25)` outputs

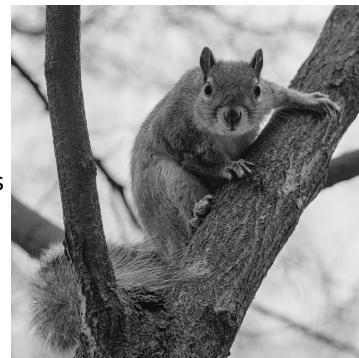


`![squirrel](./contributing/squirrel.png#w75)` outputs

**inline**

With the exception of icons, images are included as block elements by default. You can override this by including `#inline` after the image name. This can be combined with the width setting as follows.

`![squirrel](./contributing/squirrel.png#w25#inline)` outputs

**default**

By default images are presented as block elements with 100% width. So `![squirrel](./contributing/squirrel.png#w100)` and `![squirrel](./contributing/squirrel.png)` are equivalent and both output the following:





## 12.11. translating dtdocs

Translation of the darktable documentation is done via our [Weblate instance](#) .

You can either use Weblate's web UI to translate the documentation or download the translation from Weblate to your computer, edit it, and then upload the changes.

Please do all translation work through Weblate. We will not accept pull requests directly on github to update PO files.

## Making a new branch in git

1. Make a new branch to work on it in git. For example: `git checkout -b fr-translation-init`

## Adding a new language to Hugo

1. In the files `config.yaml` and `config-pdf.yaml`, locate the `languages:` line.
2. Add the language you wish to translate. For example, the English looks like this:

```
en-us:  
  title: darktable 3.4 user manual  
  weight: 1
```

3. Save the files.

## Generating a PO file

Do the following steps if you want to update the POT and PO files from the markdown source.

1. Create an empty PO file for your language in the `po` folder with the file name `content.<language>.po`. For example:  
`touch po/content.fr-fr.po`
2. Run the script to populate the PO file: `cd tools/ && ./generate-translations.sh --no-translations`

## Generating translated files

Do the following steps to generate the website files from a translation.

1. Generate the translated files: `cd tools/ && ./generate-translations.sh --no-update.`
2. Check the translation by starting hugo's internal server: `hugo server`
3. Open a web browser and check the changes. The URL is in the output of the `hugo server` command.
4. Remove the translated files, as we never check them into git: `cd tools/ && ./generate-translations.sh --rm-translations.`

## Translating website, epub, and PDF strings

There are two themes for the darktable documentation: one for the HTML website and one for the PDF. You'll need to translate the strings for both.

1. Go to `themes/hugo-darktable-docs-themes/i18n`.
2. Copy content of the file `en.yaml` and name the new file `<your language>.yaml`.
3. Translate the content of the new yaml file.
4. Check the translated PO file into git, push it to github, and open a pull request to have your changes accepted.
5. Repeat the last four steps for the other themes, `themes/hugo-darktable-docs-epub-theme` and `themes/hugo-darktable-docs-pdf-theme`.

# Integrating new translations from Weblate

The following assumes that you're git working directory is clean, that you have API access to the Weblate instance, that you've configured the Weblate git repo as a remote in your local dtdocs git repo, and that `wlc`, the Weblate command line client, is configured.

1. Commit any changes in Weblate: `wlc commit darktable/dtdocs`
2. Lock the Weblate project to prevent further changes: `wlc lock darktable/dtdocs`
3. In your local dtdocs git repo, create a new branch: `git checkout -b po-updates`
4. Update the Weblate remote: `git remote update weblate`
5. Merge the Weblate changes into your locally created branch: `git merge weblate/master`
6. Squash all the Weblate commits, since there are so many: `git reset $(git merge-base master $(git rev-parse --abbrev-ref HEAD))`
7. Stage the changed PO files: `git add -A`
8. Commit the PO files: `git commit -m "Updated with the PO files from weblate."`
9. Update the POT and PO files: `cd tools/ && ./generate-translations.sh --no-translations && cd ..`
10. Stage the POT and PO files: `git add -A`
11. Commit the POT and PO files: `git commit -m "Updated POT and PO files."`
12. Create a Pull Request in Github.
13. After the Pull Request is accepted, reset the Weblate repo to match the dtdocs repo: `wlc reset darktable/dtdocs`

darktable 3.8 user manual  
February 22, 2024