

Lua scripts

Description

darktable can be customized and extended using the Lua programming language. This repository contains the collected efforts of the darktable developers, maintainers, contributors and community. The following sections list the scripts contained in the repository, whether they can be run by themselves (Standalone - Yes) or depend on other scripts (Standalone - No), what operating systems they are known to work on (L - Linux, M - MacOS, W - Windows), and their purpose.

For the latest changes, see the [ChangeLog](#)

Official Scripts

These scripts are written primarily by the darktable developers and maintained by the authors and/or repository maintainers. They are located in the `official/` directory.

Name	Standalone	OS	Purpose
check_for_updates	Yes	LMW	Check for updates to darktable
copy_paste_metadata	Yes	LMW	Copy and paste metadata, tags, ratings, and color labels between images
delete_long_tags	Yes	LMW	Delete all tags longer than a specified length
delete_unused_tags	Yes	LMW	Delete tags that have no associated images
enfuse	No	L	Exposure blend several images (HDR)
generate_image_txt	No	L	Generate txt sidecar files to be overlaid on zoomed images
image_path_in_ui	Yes	LMW	Plugin to display selected image path

Name	Standalone	OS	Purpose
import_filter_manager	Yes	LMW	Manager for import filters
import_filters	No	LMW	Two import filters for use with import_filter_manager
save_selection	Yes	LMW	Provide save and restore from multiple selection buffers
selection_to_pdf	No	L	Generate a PDF file from the selected images

Contributed Scripts

These scripts are contributed by users. They are meant to have an "owner", i.e. the author, who maintains them. Over time the community has helped maintain these scripts, as well as the authors. They are located in the contrib/ directory.

Name	Standalone	OS	Purpose
AutoGrouper	Yes	LMW	Group images together by time
autostyle	Yes	LMW	Automatically apply styles on import
change_group_leader	Yes	LMW	Change which image leads group
clear_GPS	Yes	LMW	Reset GPS information for selected images
CollectHelper	Yes	LMW	Add buttons to selected images module to manipulate the collection
copy_attach_detach_tags	Yes	LMW	Copy and paste tags from/to images
cr2hdr	Yes	L	Process image created with Magic Lantern Dual ISO
enfuseAdvanced	No	LMW	Merge multiple images into Dynamic Range Increase (DRI) or Depth From Focus (DFF) images
exportLUT	Yes	LMW	Create a LUT from a style and export it
ext_editor	No	LW	Export pictures to collection and edit them with up to nine user-defined external editors
face_recognition	No	LM	Identify and tag images using facial recognition
fujifilm_dynamic_range	No	LMW	Correct fujifilm exposure based on exposure bias camera setting
fujifilm_ratings	No	LM	Support importing Fujifilm ratings
geoJSON_export	No	L	Create a geo JSON script with thumbnails for use in ...
geoToolbox	No	LMW	A toolbox of geo functions
gimp	No	LMW	Open an image in GIMP for editing and return the result

Name	Standalone	OS	Purpose
gpx_export	No	LMW	Export a GPX track file from selected images GPS data
HDRMerge	No	LMW	Combine the selected images into an HDR DNG and return the result
hugin	No	LMW	Combine selected images into a panorama and return the result
image_stack	No	LMW	Combine a stack of images to remove noise or transient objects
image_time	Yes	LMW	Adjust the EXIF image time
kml_export	No	L	Export photos with a KML file for usage in Google Earth
LabelsToTags	Yes	LMW	Apply tags based on color labels and ratings
OpenInExplorer	No	LMW	Open the selected images in the system file manager
passport_guide	Yes	LMW	Add passport cropping guide to darkroom crop tool
pdf_slideshow	No	LM	Export images to a PDF slideshow
photils	No	LM	Automatic tag suggestions for your images
quicktag	Yes	LMW	Create shortcuts for quickly applying tags
rate_group	Yes	LMW	Apply or remove a star rating from grouped images
rename_images	Yes	LMW	Rename single or multiple images
rename-tags	Yes	LMW	Change a tag name
RL_out_sharp	No	LW	Output sharpening using GMic (Richardson-Lucy algorithm)
select_untagged	Yes	LMW	Enable selection of untagged images
slideshowMusic	No	L	Play music during a slideshow
transfer_hierarchy	Yes	LMW	Image move/copy preserving directory hierarchy
video_ffmpeg	No	LMW	Export video from darktable

Example Scripts

These scripts provide examples of how to use specific portions of the API. They run, but are meant for demonstration purposes only. They are located in the examples/ directory.

Name	Standalone	OS	Purpose
api_version	Yes	LMW	Print the current API version

Name	Standalone	OS	Purpose
darkroom_demo	Yes	LMW	Demonstrate changing images in darkroom
gettextExample	Yes	LM	How to use translation
hello_world	Yes	LMW	Prints hello world when darktable starts
lighttable_demo	Yes	LMW	Demonstrate controlling lighttable mode, zoom, sorting and filtering
moduleExample	Yes	LMW	How to create a lighttable module
multi_os	No	LMW	How to create a cross platform script that calls an external executable
panels_demo	Yes	LMW	Demonstrate hiding and showing darktable panels
preferenceExamples	Yes	LMW	How to use preferences in a script
printExamples	Yes	LMW	How to use various print functions from a script
running_os	Yes	LMW	Print out the running operating system

Tools

Tool scripts perform functions relating to the repository, such as generating documentation. They are located in the tools/ directory.

Name	Standalone	OS	Purpose
executable_manager	Yes	LMW	Manage the external executables used by the lua scripts
gen_i18n_mo	No	LMW	Generate compiled translation files (.mo) from source files (.po)
get_lib_manpages	No	LM	Retrieve the library documentation and output it in man page and PDF format
get_libdoc	No	LMW	Retrieve the library documentation and output it as text
script_manager	No	LMW	Manage (install, update, enable, disable) the lua scripts

Related third-party projects

The following third-party projects are listed for information only. Think of this collection as an `awesome-darktable-lua-scripts` list. Use at your own risk!

- [trougnouf/dtMediaWiki](#) – Wikimedia Commons export
- [wpferguson/extra-dt-lua-scripts](#)
- [xxv/darktable-git-annex](#) – git-annex integration
- [theres/dt_backup](#) – automatic backup on exit
- [Sitwon/dt_fujifilm_ratings](#) – Fujifilm Ratings import
- [progo/leica-q-autocrop](#) – crop in 35 mm or 50 mm equivalent

- BzKevin/OpenInPS-Darktable-PlugIn – open in Adobe Photoshop
- johnnyrun/darktable_lua_gimp – GIMP export
- arru/darktable-scripts
- nbremond77/darktable
- s5k6/dtscripts
- ChristianBirzer/darktable_extra_scripts

Download and Install

The recommended method of installation is using git to clone the repository. This ensures that all dependencies on other scripts are met as well as providing an easy update path. Single scripts listed as standalone may be downloaded and installed by themselves.

snap packages

The snap version of darktable comes with lua included starting with version 2.4.3snap2.

Ensure git is installed on your system. If it isn't, use the package manager to install it. Then open a terminal and:

```
cd ~/snap/darktable/current
git clone https://github.com/darktable-org/lua-scripts.git lua
```



flatpak packages

Flatpak packages now use the internal lua interpreter.

Ensure git is installed on your system. If it isn't, use the package manager to install it. Then open a terminal and:

```
cd ~/.var/app/org.darktable.Darktable/config/darktable
git clone https://github.com/darktable-org/lua-scripts.git lua
```



appimage packages

These packages run in their own environment and don't have access to a lua interpreter, therefore the scripts can't run. The packagers could enable the internal interpreter, or allow the package to link the interpreter from the operating system, or bundle a copy of lua with the package. If you use one of these packages and wish to use the lua scripts, please contact the package maintainer and suggest the above fixes.

Linux and MacOS

Ensure git is installed on your system. If it isn't, use the package manager to install it. Then open a terminal and:

```
cd ~/.config/darktable/  
git clone https://github.com/darktable-org/luascripts.git lua
```



Windows

Ensure git is installed on your system. Git can be obtained from <https://gitforwindows.org/>, as well as other places. If you use the gitforwindows.org distribution, install the Git Bash Shell also as it will aid in debugging the scripts if necessary. Then open a command prompt and run:

```
cd %LOCALAPPDATA%\darktable  
git clone https://github.com/darktable-org/luascripts.git lua
```



If you don't have %LOCALAPPDATA%\darktable you have to start darktable at least once, because the directory is created at the first start of darktable.

Enabling

When darktable starts it looks for a file name `~/.config/darktable/luarc` (`%LOCALAPPDATA%\darktable\luarc` for windows) and reads it to see which scripts to include. The file is a plain text file with entries of the form `require "<directory>/<name>"` where `directory` is the directory containing the scripts, from the above list, and `name` is the name from the above list. To include GIMP the line would be `require "contrib/gimp" .`

The recommended way to enable and disable specific scripts is using the script manager module. To use script manager do the following:

Linux or MacOS

```
echo 'require "tools/script_manager"' > ~/.config/darktable/luarc
```



Windows

```
echo require "tools/script_manager" > %LOCALAPPDATA%\darktable\luarc
```



Snap

```
echo 'require "tools/script_manager"' > ~/snap/darktable/current/luarc
```



Flatpak

```
echo 'require "tools/script_manager"' >  
~/var/app/org.darktable.Darktable/config/darktable/luarc
```



You can also create or add lines to the luarc file from the command line:

```
echo 'require "contrib/gimp"' > ~/.config/darktable/luarc to create the file with a gimp entry  
or echo 'require "contrib/hugin"' >> ~/.config/darktable/luarc to add an entry for hugin.
```

On windows from a command prompt:

```
echo require "contrib/gimp" > %LOCALAPPDATA%\darktable\luarc to create the file with a gimp entry  
or echo require "contrib/hugin" >> %LOCALAPPDATA%\darktable\luarc to add an entry for hugin.
```

Disabling

To disable a script open the luarc file in your text editor and insert `--` at the start of the line containing the script you wish to disable, then save the file.

Updating

To update the script repository, open a terminal or command prompt and do the following:

Snap

```
cd ~/snap/darktable/current/luarc  
git pull
```



Flatpak

```
cd ~/.var/app/org.darktable.Darktable/config/darktable/luarc  
git pull
```



Linux and MacOS

```
cd ~/.config/darktable/luarc  
git pull
```



Windows

```
cd %LOCALAPPDATA%\darktable\luarc  
git pull
```

Documentation

The Lua Scripts Manual provides documentation for the scripts transcribed from the header comments. Each script also contains comments and usage instructions in the header comments.

The Lua Scripts Library API Manual provides documentation of the libraries and functions. Lua-script libraries documentation may also be generated using the tools in the tools/ directory.

More information about the scripting with Lua can be found in the darktable user manual: Scripting with Lua

The Lua API Manual provides documentation of the darktable Lua API.

Troubleshooting

Running darktable with Lua debugging enabled provides more information about what is occurring within the scripts.

Snap

Open a terminal and start darktable with the command `snap run darktable -d lua`. This provides debugging information to give you insight into what is happening.

Linux

Open a terminal and start darktable with the command `darktable -d lua`. This provides debugging information to give you insight into what is happening.

MacOS

Open a terminal and start darktable with the command `/Applications/darktable.app/Contents/MacOS/darktable -d lua`. This provides debugging information to give you insight into what is happening.


Windows

Open a command prompt. Start darktable with the command `"C:\Program Files\darktable\bin\darktable" -d lua > log.txt`. This provides debugging information to give you insight into what is happening.

Contributing

In order to have your own scripts added here they have to be under a free license (GPL2+ will definitely work, others can be discussed).

Releases

 1 tags

Packages

No packages published

Contributors 44



+ 33 contributors

Languages

- Lua 100.0%