Building darktable 2.2

**NOTE:** This is a work in progress, currently contains a lot of outdated entries from 2.0 build instructions.

**General build instructions**

First, install git, clone the repo and check out the latest stable release:

```bash
git clone https://github.com/darktable-org/darktable.git
cd darktable
git tag # find the latest 2.2 release, e.g. release-2.2.0
git checkout release-2.2.0
```

Then install the dependencies (listed below) and build:

```
./build.sh
```

To build the usermanual:

```bash
cd build
make darktable-usermanual
```

To additionally build the lua-api manual

```
make darktable-lua-api
```

**Ubuntu 15.10**

**Note:** including this in order to support those with AMD graphics cards that require the fglrx driver - no longer supported on Ubuntu 16.04 or above.

**Note:** There is no llvm-3.9/clang-3.9 in Ubuntu 16.04, so test-compilation of OpenCL programs can not be done.

- Build status: **PASS**

Perform the instructions for Ubuntu 16.04, plus:

Create CLI script for jsonschema:

```bash
echo "#!/ /usr/bin/python3
# EASY-INSTALL-ENTRY-SCRIPT: 'jsonschema==2.4.0','console_scripts','jsonschema'
__requires__ = 'jsonschema==2.4.0'
import sys
from pkg_resources import load_entry_point
if __name__ == '__main__':
    sys.exit(1)
    load_entry_point('jsonschema==2.4.0', 'console_scripts', 'jsonschema')()
" | sudo tee /usr/local/bin/jsonschema
sudo chmod +x /usr/local/bin/jsonschema
```

**Ubuntu 16.04**

**Note:** binary packages are available from [pmdebruijn's release ppa](https://ppa.launchpad.net/pmdebruijn/ppa).

- Build status: **PASS**

**Note:** There is no llvm-3.9/clang-3.9 in Ubuntu 16.04, so test-compilation of OpenCL programs can not be done.

**Minimal dependencies:**
sudo apt-get install gcc g++ cmake intltool xsltproc libgtk-3-dev libxml2-utils libxml2-dev libblen
sfun-dev librsvg2-dev libsqlite3-dev libcurl4-gnutls-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpng-dev libpugixml-dev python3-pkg-resources

Optional dependencies:

sudo apt-get install libgphoto2-dev libgsoup2.4-dev libopenexr-dev libwebp-dev libflickcurl-dev libopenjpeg-dev libsecret-1-dev libgraphicsmagick1-dev libcolorord-dev libcolorord-gtk-dev libcurl-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev libsqlite3-dev libcurl4-gnutls-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev python3-pkg-resources

Usermanual:

sudo apt-get install default-jdk gnome-doc-utils libxml2-dev libexiv2-dev libpng-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev python3-pkg-resources

Translated man pages:

sudo apt-get install po4a

Ubuntu 16.10

Note: binary packages are available from pmidebruijn's release ppa.

* Build status: PASS

Minimal dependencies:

sudo apt-get install gcc g++ cmake intltool xsltproc libgtk-3-dev libxml2-utils libxml2-dev libblen

Optional dependencies:

sudo apt-get install libgphoto2-dev libgsoup2.4-dev libopenexr-dev libwebp-dev libflickcurl-dev libopenjpeg-dev libsecret-1-dev libgraphicsmagick1-dev libcolorord-dev libcolorord-gtk-dev libcurl-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev python3-pkg-resources

Usermanual:

sudo apt-get install default-jdk gnome-doc-utils libxml2-dev libexiv2-dev libpng-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev python3-pkg-resources

Translated man pages:

sudo apt-get install po4a

Ubuntu 17.04 and 17.10

Note: binary packages are available from pmidebruijn's release ppa.

* Build status: PASS

Minimal dependencies:

sudo apt-get install gcc g++ cmake intltool xsltproc libgtk-3-dev libxml2-utils libxml2-dev libblen

Optional dependencies:

export LLVM_DIR=/usr/lib/llvm-3.9/lib/cmake/llvm

Usermanual:

sudo apt-get install default-jdk gnome-doc-utils libxml2-dev libexiv2-dev libpng-dev libjpeg-dev libriff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev python3-pkg-resources

Translated man pages:

sudo apt-get install po4a
sudo apt-get install libgphoto2-dev libsoup2.4-dev libopenexr-dev libwebp-dev libflickrurl-dev libopenjp2-7-dev libsecret-1-dev libgraphicsmagic1-dev libcolorord-dev libcolorord-gtk-dev libcups2-dev libbsd11.2-dev libbsd11-image1.2-dev libbl11-mesa-dev libosmgpsmap-1.0-dev libopenjp2-7-dev llvm-3.9 clang-3.9 python3-jsonschema

Usermanual:

sudo apt-get install default-jdk gnome-doc-utils libsaxon-java fop imagemagick docbook-xml docbook-xsl

Translated man pages:

sudo apt-get install po4a

Fedora 24 and 25

Build status 24: PASS
Build status 25: PASS

Minimal dependencies:

sudo dnf install cmake gcc-c++ intltool gtk3-devel libxml2-devel lensfun-devel librsvg2-devel sqlite-devel libcurl-devel libjpeg-turbo-devel libtiff-devel lcms2-devel json-glib-devel expat-devel pugixml-devel libxslt

Optional dependencies:

Note: LLVM 3.9+ is not available on Fedora, so test-compilation of OpenCL programs can not be done.


Create CLI script for jsonschema:

```python
# EASY-INSTALL-ENTRY-SCRIPT: 'jsonschema==2.5.1','console_scripts','jsonschema'
__requires__ = 'jsonschema==2.5.1'
import sys
from pkg_resources import load_entry_point
if __name__ == '__main__':
sys.exit(load_entry_point('jsonschema==2.5.1', 'console_scripts', 'jsonschema')())
```

| sudo chmod +x /usr/local/bin/jsonschema

Usermanual:

sudo dnf install saxon ImageMagick gnome-doc-utils fop docbook-dtds docbook-style-xsl java-1.8.0-openjdk-devel

Translated man pages:

sudo dnf install po4a

openSUSE

Binary packages are available for openSUSE and SLE 12.

Build status (Leap 42.1): UNKNOWN
Build status (Leap 42.2): UNKNOWN
Build status (Tumbleweed 2016-12-??): UNKNOWN

Minimal dependencies
sudo zypper install cmake make gcc gcc-c++ intltool libxslt-tools gtk3-devel libxml2-devel libxml2-tools lensfun-devel librsvg-devel sqlite3-devel libcurl-devel libjpeg8-devel libtiff-devel liblcm s2-devel json-glib-devel libexiv2-devel pugixml-devel

Optional dependencies

sudo zypper install libgphoto2-devel openexr-devel libwebp-devel libflickrurl-devel openjpeg-devel libsecret-devel GraphicsMagick-devel libcolord-devel libcolorord-gtk-devel cups-devel libSDL-devel libSDL_image-devel libosmgpsmap-devel

Usermanual

sudo zypper install java-1_8_0-openjdk-devel gnome-doc-utils saxon6-scripts saxon6-fop imagemagick docbook_4 docbook-xsl-stylesheets

Make cmake find saxon:

cd /usr/share/java
sudo ln -s saxon6.jar saxon.jar

Translated man pages

sudo zypper install po4a xml2po

Debian 8 Jessie

Build status: PASS

Note: no sudo by default on debian; use su to go to root before installing.

Minimal dependencies

apt-get install gcc g++ cmake intltool xsltproc libgtk-3-dev libxml2-utils libxml2-dev liblensfun-dev librsvg2-dev libsqlite3-dev libcurl4-gnutls-dev libjpeg-dev libtiff5-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev

Optional dependencies

apt-get install libgphoto2-dev libsoup2.4-dev libopenexr-dev libwebp-dev libflickrurl-dev desktop-file-utils libopenjpeg-dev libsecret-1-dev libgraphicsmagick1-dev libcolorord-dev libcolorord-gtk-dev libcups2-dev libdsdl1.2-dev libdsd-imagel.2-dev libgl1-mesa-dev libosmgpsmap-1.0-0-dev libopenjp2-7-dev echo \"deb http://ftp.debian.org/debian jessie-backports main\" > /etc/apt/sources.list.d/backports.list apt-get update apt-get -t jessie-backports install python3-jsonschema

OpenCL test compilation

Unable to do OpenCL test compilation as clang-3.9 libclang-common-3.9-dev llvm-3.9-dev packages aren't available in Jessie.

Usermanual:

apt-get install default-jdk gnome-doc-utils libsaxon-java fop imagemagick docbook-xml docbook-xsl

Translated man pages:

apt-get install po4a

Arch Linux

Build status: UNKNOWN

There is a package in the arch user repository (AUR) called darktable-git. At the time of writing this was rather out of date.

Minimal dependencies
sudo pacman -S base-devel cmake intltool lensfun curl exiv2 lcms2 librsvg libxslt sqlite

mkdir aur
cd aur/
git clone https://aur.archlinux.org/pugixml.git
cd pugixml/
makepkg -sri
cd

Optional dependencies

sudo pacman -S openexr libwebp flickcurl graphicsmagick libcups libgphoto2 sdl mesa-libgl
dbus-glib

cd aur
git clone https://aur.archlinux.org/osm-gps-map.git
cd osm-gps-map
makepkg -sri
cd

Usermanual

sudo pacman -S jdk8-openjdk gnome-doc-utils fop imagemagick extra/docbook-xml extra/docbook-xsl

cd aur
git clone https://aur.archlinux.org/saxon6.git
cd saxon6/
makepkg -sri
cd /usr/share/java
ln -s saxon6/saxon.jar
echo '#!/bin/sh
exec java -classpath /usr/share/java/saxon.jar com.icl.saxon.StyleSheet "@0"' | sudo tee /usr/local/bin/saxon-xslt
cd

Translated man pages

sudo pacman -S po4a

Linux Mint 18

Build status: UNKNOWN

Minimal dependencies:

sudo apt-get install gcc g++ cmake intltool xsltproc libgtk-3-dev libxml2-utils libxml2-dev libxen
sfun-dev librsvg2-dev libsqlite3-dev libcurl4-gnutls-dev libjpeg-dev libtiff4-dev liblcms2-dev libjson-glib-dev libexiv2-dev libpugixml-dev

Optional dependencies:

sudo apt-get install libgphoto2-dev libglesp2.4-dev libopenexr-dev libwebp-dev libflickcurl-dev libopenjpeg-dev libsecret-1-dev graphicsmagick1-dev libcolorord-dev libcolorord-gtk-dev libglesp2-dev libexpat1-dev libodium1.2-dev libodlm1-mesa-dev

Note that osmogpsmap 1.0.2 does not exist in the mint package archive. We can use the same resolution to the problem in Ubuntu 14.04 above by adding the ubuntu trusty-backports source and installing from there. Update the country code in the deb source below (or your favourite mirror):

echo 'deb http://au.archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multivers
e' | sudo tee -a /etc/apt/sources.list.d/additional-repositories.list
echo 'Package: *
Pin: release a=trustly-backports
Pin-Priority: 100' | sudo tee -a /etc/apt/preferences
sudo apt-get update
sudo apt-get install libosmgpsmap-1.0-dev

Create the missing pkg-config file:

```
# Generated using the following data

prefix=/usr
exec_prefix=${prefix}
libdir=${prefix}/lib/x86_64-linux-gnu
includedir=${prefix}/include

Name: osm-gps-map
Description: Moving map widget using openstreet map data
Version: 1.0.2
Requires: gtk+-3.0 libsoup-2.4
Libs: -L${libdir} -losmgpsmap-1.0
Cflags: -I${includedir}/osmgpsmap-1.0' | sudo tee /usr/lib/x86_64-linux-gnu/pkgconfig/osmgpsmap-1.0.pc
```

**Usermanual:**

```
sudo apt-get install default-jdk gnome-doc-utils libsaxon-java fop imagemagick docbook-xml docbook-xsl
```

**Translated man pages:**

```
sudo apt-get install po4a
```

**Gentoo Linux**

Build status: **UNKNOWN**

- building usermanual appeared to hang at:

```bash
Scanning dependencies of target darktable_single_xml
[ 96%] Generating the profiled docbook xml file
```

But completed 34 minutes later!

**Note:** need sse3 USE flag to be set by adding sse3 to /etc/portage/make.conf, e.g.

```
USE="bindist mmx sse sse2 sse3"
```

**Rebuild:**

```
emerge --update --deep --newuse @world
emerge --depclean
```

**Minimal dependencies:**

```
```

**Optional dependencies:**

```
emerge --ask media-libs/libgphoto2 net-libs/libsoup media-libs/openercx media-libs/libwebp media-libs/flickr curl media-libs/openjpeg app-crypt/libsecret media-gfx/graphicsmagick x11-misc/coloro x11-libs/coloro-gtk net-print/cups media-libs/libsdl media-libs/sdl-image media-libs/mesa
```

**Optional dependencies that are missing from the package archive:**

- sci-geosciences/osm-gps-map which is present in the gentoo archives is too old. Build from source:

```
emerge --ask dev-libs/gobject-introspection x11-libs/cairo gnome-base/gnome-common dev-util/gtk-doc
git clone git://github.com/nzjrs/osm-gps-map
```

04/01/2020
cd osm-gps-map
./autogen.sh
make
su
make install
export PKG_CONFIG_PATH=/usr/local/lib64/pkgconfig:$PKG_CONFIG_PATH
echo 'export PKG_CONFIG_PATH=/usr/local/lib64/pkgconfig:$PKG_CONFIG_PATH' >> ~/.bash_profile

- media-libs/openjpeg is version 2.1.0, need 1.x, so we build from source:

git clone https://github.com/uclouvain/openjpeg.git
cd openjpeg
git tag # find latest 1.x tag
git checkout version.1.5.2 # latest tag from previous line
cmake
make
su
mkdir /usr/local/share/pkgconfig/
make install

Usermanual:

For one of the dependencies of fop (dev-java/sun-jms-1.1-r2), we need to do this:

Fetch instructions for dev-java/sun-jms-1.1-r2:

*  
  * Due to license restrictions, we cannot fetch the distributables automagically.
  *  
  * 2. Accept the License Agreement
  * 3. Download jms-1_1-fr-apidocs.zip
  * 4. Move the file to /usr/portage/distfiles

Note that after copying the zipfile into /usr/portage/distfiles/, I needed to do the following, or the emerge would fail:

su
chmod a+r /usr/portage/distfiles/jms-1_1-fr-apidocs.zip

Then:

emerge --ask virtual/jdk app-text/gnome-doc-utils dev-java/fop media-gfx/imagemagick app-text/docbook-xslt app-text/docbook-xsl-stylesheets

Note: Saxon 6.5 is not present in the Gentoo package archive (the dev-java/saxon package will give you saxon-he 9.3 rather than saxon 6.5 as required by darktable). To install saxon 6.5:

- download the saxon 6.5 zip file from http://saxon.sourceforge.net/#F6.5.5.

su
mkdir -p /usr/local/share/java/
unzip saxon6-5-5.zip -d /usr/local/share/java/saxon6-5-5
mkdir -p /usr/share/java
cd /usr/share/java
ln -s /usr/local/share/java/saxon6-5-5/saxon.jar
ln -s /usr/local/share/java/saxon6-5-5/saxon-xml-apis.jar
echo '#!/bin/sh
exec java -classpath /usr/share/java/saxon.jar com.icl.saxon.StyleSheet "@0";'> /usr/local/bin/saxon-xslt
chmod a+x /usr/local/bin/saxon-xslt

Translated man pages:

emerge --ask app-text/po4a
Manjaro Linux 18.0

Build status: Testing

In addition to the standard installation, the following packages have to be installed.

**Minimal dependencies**

```
sudo pacman -S cmake intltool libcurl-gnutls intltool llvm6 llvm6-libs clang
```

**Optional dependencies**

In the standard installation of Manjaro all optional packages are already installed for darktable.

**Usermanual**

```
sudo pacman -S gnome-doc-utils fop docbook-xml docbook-xsl
```

Additional saxon6 is required but the installation from AUR does not work and saxon6 has to be installed manually.

First download the saxon 6.5 zip file from [http://saxon.sourceforge.net/#F6.5.5](http://saxon.sourceforge.net/#F6.5.5) and switch to the folder with the downloaded zip file.

Now run the following commands:

```
sudo mkdir -p /usr/local/share/java/
sudo unzip saxon6-5-5.zip -d /usr/local/share/java/saxon6-5-5
cd /usr/local/share/java
sudo ln -s /usr/local/share/java/saxon6-5-5/saxon.jar
sudo ln -s /usr/local/share/java/saxon6-5-5/saxon-xm-apis.jar
echo '#!/bin/sh' | sudo tee /usr/local/bin/saxon-xslt
echo '' | sudo tee -a /usr/local/bin/saxon-xslt
echo 'exec java -classpath /usr/share/java/saxon.jar com.icl.saxon.StyleSheet "@0"' | sudo tee -a /usr/local/bin/saxon-xslt
sudo chmod a+x /usr/local/bin/saxon-xslt
```

Finally Oracle Java must be installed from AUR and set as default.

```
sudo yaourt jdk
```

Check which jdk has been installed in your system.

```
archlinux-java status
```

Now set Oracle Java as the default jdk.

```
sudo archlinux-java set java-11-jdk
```

**Translated man pages**

```
sudo pacman -S po4a
```