# **Build Instructions For Linux**

Debian-based Linux distributions and FreeBSD can use automatic build scripts supplied by the project, which can handle dependency installation, building and packaging. Additional build directions are available for other distributions below.

****Note:**** as of May 1, 2019, [Facebook live now mandates the use of RTMPS](https://developers.facebook.com/docs/graph-api/changelog/breaking-changes/" \l "live-api-4-24). That functionality requires your distro's [mbed TLS](https://tls.mbed.org/) package, which [obs-studio/cmake/Modules/FindMbedTLS.cmake script](https://github.com/obsproject/obs-studio/blob/master/cmake/Modules/FindMbedTLS.cmake) searches for at compile time.

****Note:**** Do not use the GitHub source .tar as it does not include all the required source files. Always use the appropriate Git tag with the associated submodules.

****Note:**** OBS will not run by default from within the build directory, throwing an error similar to "error while loading shared libraries: libobs-frontend-api.so.0: cannot open shared object file: No such file or directory". Use the portable mode options for running the binaries without doing a system install.

# **Debian-based build directions**

## Option A: Automatic Debian builds

Automatic Debian builds allow building OBS with minimal input and setup - necessary dependencies are installed automatically, build flags use a sane default and the generated OBS build uses the application's full feature set.

### Prerequisites

* Debian Bullseye
* Git

### Build procedure

Clone the repository including ****submodules****:

git clone --recursive https://github.com/obsproject/obs-studio.git

To do a ****fully automated**** build, open Terminal, switch to the checkout directory then run one of the following commands:

# Download and set up dependencies, then build OBS for local host

# architecture with common feature set

CI/build-linux.sh

# Skip download and setup of dependencies

CI/build-linux.sh --skip-dependency-checks

# Show all available options

CI/build-linux.sh --help

# Use `my\_build\_dir` prefix as build directory

CI/build-linux.sh --build-dir my\_build\_dir

## Option B: Custom Debian builds

Custom Debian builds allow full customization of the desired build configuration but also require manual setup and preparation. Available CMake configuration variables can be found in the [CMake build system documentation](https://obsproject.com/wiki/building-obs-studio" \l "cmake).

### Prerequisites

* Debian Bullseye
* CMake 3.16 or newer
* Git
* Ninja
* *Optional:* CCache to improve compilation speeds on consecutive builds
* For browser source and browser panel support, the pre-built CEF framework is needed:
  + Chromium Embedded Framework (CEF) [x86\_64](https://cdn-fastly.obsproject.com/downloads/cef_binary_5060_linux64.tar.bz2)
* Several additional dependencies (see step 2 below)

### Build procedure

#### **1. Get the source code**

1. Open a Terminal window, create and switch to a directory you want to have OBS checked out
2. Clone the repository including ****submodules****: git clone --recursive https://github.com/obsproject/obs-studio.git

(If you do not know what submodules are, or you are not using Git from the command line, ****PLEASE make sure to fetch the submodules too****.)

#### **2. Get the dependencies**

* To download and set up most preconditions mentioned above, you can also run the script CI/linux/01\_install\_dependencies.sh from the checkout directory (run it with the --help switch to see all available options).

****NOTE:**** The directory where the script will download and setup the dependencies in cannot be changed.

Alternatively the required dependencies can be installed using apt:

Build system dependencies

sudo apt install cmake ninja-build pkg-config clang clang-format build-essential curl ccache git

OBS dependencies (core):

sudo apt install libavcodec-dev libavdevice-dev libavfilter-dev libavformat-dev libavutil-dev libswresample-dev libswscale-dev libx264-dev libcurl4-openssl-dev libmbedtls-dev libgl1-mesa-dev libjansson-dev libluajit-5.1-dev python3-dev libx11-dev libxcb-randr0-dev libxcb-shm0-dev libxcb-xinerama0-dev libxcb-composite0-dev libxcomposite-dev libxinerama-dev libxcb1-dev libx11-xcb-dev libxcb-xfixes0-dev swig libcmocka-dev libxss-dev libglvnd-dev libgles2-mesa libgles2-mesa-dev libwayland-dev librist-dev libsrt-openssl-dev libpci-dev

OBS dependencies (UI):

If Qt6 is not available:

sudo apt install qtbase5-dev qtbase5-private-dev libqt5svg5-dev qtwayland5

Plugin dependencies:

sudo apt install libasound2-dev libfdk-aac-dev libfontconfig-dev libfreetype6-dev libjack-jackd2-dev libpulse-dev libsndio-dev libspeexdsp-dev libudev-dev libv4l-dev libva-dev libvlc-dev libdrm-dev

#### **3. Set up the build project**

1. Run CMake to generate a build environment

cmake -S . -B YOUR\_BUILD\_DIRECTORY -G Ninja \

-DCEF\_ROOT\_DIR="../obs-build-dependencies/cef\_binary\_5060\_linux64" \

-DENABLE\_PIPEWIRE=OFF \

-DENABLE\_AJA=0

****Optional Settings:****

1. To enable PipeWire support change -DENABLE\_PIPEWIRE to ON
2. To disable browser source support (e.g. for 32-bit builds) set -DENABLE\_BROWSER to OFF
3. To change the build type, pass either Debug, Release, RelWithDebInfo, or MinSizeRel as -DCMAKE\_BUILD\_TYPE

****NOTE:**** When building OBS with LINUX\_PORTABLE disabled, OBS expects GNU-based install paths (e.g. /usr/local/[bin,lib,share]) and is built for a single architecture only. To create separate builds for 32-bit and 64-bit architectures, always enable portable builds.

#### **4. Build the project**

1. Run cmake --build YOUR\_BUILD\_DIRECTORY to build the entire OBS project
2. Run cmake --build YOUR\_BUILD\_DIRECTORY -t libobs to build only libobs or any other valid target
3. Run cmake --build YOUR\_BUILD\_DIRECTORY -t clean to clean your current build directory

#### **6. Install the project**

Installation will use the directory specified via -DCMAKE\_INSTALL\_PREFIX or can be customised with the --prefix switch:

1. Run cmake --install YOUR\_BUILD\_DIRECTORY to install OBS to the prefix the project was configured with
2. Run cmake --install YOUR\_BUILD\_DIRECTORY --prefix to install OBS to a custom location

#### **7. Create Debian package**

1. Run cmake --build YOUR\_BUILD\_DIRECTORY --target package - CMake will handle all operations necessary to create a .deb package archive, including necessary dependencies.

# **Red Hat-based**

Get RPM Fusion at <http://rpmfusion.org/Configuration/> ([Nux Desktop](http://li.nux.ro/repos.html) is an alternative that may include better packages for RHEL/CentOS 7)

Get the required packages:

sudo yum install \

alsa-lib-devel \

asio-devel \

cmake \

ffmpeg-free-devel \

fontconfig-devel \

freetype-devel \

gcc \

gcc-c++ \

gcc-objc \

git \

glib2-devel \

json-devel \

libavcodec-free-devel \

libavdevice-free-devel \

libcurl-devel \

libdrm-devel \

libglvnd-devel \

libjansson-devel \

libuuid-devel \

libva-devel \

libv4l-devel \

libX11-devel \

libXcomposite-devel \

libXinerama-devel \

luajit-devel \

make \

mbedtls-devel \

pciutils-devel \

pipewire-devel \

pulseaudio-libs-devel \

python3-devel \

qt5-qtbase-devel \

qt5-qtbase-private-devel \

qt5-qtsvg-devel \

qt5-qtwayland-devel \

qt5-qtx11extras-devel \

speexdsp-devel \

swig \

systemd-devel \

vlc-devel \

wayland-devel \

websocketpp-devel \

x264-devel

* + If speexdsp-devel is not available, it can be built from source (<https://gitlab.xiph.org/xiph/speexdsp>)

Building and installing OBS:

If building with browser source:

wget https://cdn-fastly.obsproject.com/downloads/cef\_binary\_5060\_linux64.tar.bz2

tar -xjf ./cef\_binary\_5060\_linux64.tar.bz2

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DENABLE\_BROWSER=ON -DCEF\_ROOT\_DIR="../../cef\_binary\_5060\_linux64" -DENABLE\_AJA=OFF -DENABLE\_NEW\_MPEGTS\_OUTPUT=OFF ..

make -j4

sudo make install

echo "/usr/local/lib" | sudo tee /etc/ld.so.conf.d/local.conf -a

sudo ldconfig

If building without browser source:

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DENABLE\_BROWSER=OFF -DENABLE\_AJA=OFF -DENABLE\_NEW\_MPEGTS\_OUTPUT=OFF ..

make -j4

sudo make install

echo "/usr/local/lib" | sudo tee /etc/ld.so.conf.d/local.conf -a

sudo ldconfig

By default OBS installs libraries in /usr/local/lib.

# **Fedora**

* Add the RPM Fusion repo. This is required for some packages such as vlc-devel and x264-devel.

sudo dnf install \

https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm

Get the required packages:

sudo dnf install \

alsa-lib-devel \

asio-devel \

cmake \

ffmpeg-free-devel \

fontconfig-devel \

freetype-devel \

gcc \

gcc-c++ \

gcc-objc \

git \

glib2-devel \

jansson-devel \

json-devel \

libavcodec-free-devel \

libavdevice-free-devel \

librist-devel \

libcurl-devel \

libdrm-devel \

libglvnd-devel \

libuuid-devel \

libva-devel \

libv4l-devel \

libX11-devel \

libXcomposite-devel \

libXdamage \

libXinerama-devel \

libxkbcommon-devel \

luajit-devel \

make \

mbedtls-devel \

pciutils-devel \

pipewire-devel \

pulseaudio-libs-devel \

python3-devel \

qt6-qtbase-devel \

qt6-qtbase-private-devel \

qt6-qtsvg-devel \

qt6-qtwayland-devel \

speexdsp-devel \

swig \

systemd-devel \

vlc-devel \

wayland-devel \

websocketpp-devel \

x264-devel

Building and installing OBS:

If building with browser source:

wget https://cdn-fastly.obsproject.com/downloads/cef\_binary\_5060\_linux64.tar.bz2

tar -xjf ./cef\_binary\_5060\_linux64.tar.bz2

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DENABLE\_BROWSER=ON -DCEF\_ROOT\_DIR="../../cef\_binary\_5060\_linux64" -DENABLE\_AJA=OFF -DENABLE\_NEW\_MPEGTS\_OUTPUT=OFF ..

make -j$(nproc)

sudo make install

echo "/usr/local/lib" | sudo tee /etc/ld.so.conf.d/local.conf -a

sudo ldconfig

If building without browser source:

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DENABLE\_BROWSER=OFF -DENABLE\_AJA=OFF -DENABLE\_NEW\_MPEGTS\_OUTPUT=OFF ..

make -j$(nproc)

sudo make install

echo "/usr/local/lib" | sudo tee /etc/ld.so.conf.d/local.conf -a

sudo ldconfig

# **openSUSE**

See [openSUSE install instructions](https://obsproject.com/wiki/install-instructions" \l "opensuse-unofficial) for details on adding Packman repository.

Get the required packages:

sudo zypper in cmake \

fontconfig-devel \

freetype2-devel \

gcc \

gcc-c++ \

libcurl-devel \

ffmpeg2-devel \

libjansson-devel \

libpulse-devel \

libspeexdsp-devel \

libqt5-qtbase-devel \

libqt5-qtx11extras-devel \

libudev-devel \

libv4l-devel \

libXcomposite-devel \

libXinerama-devel \

libXrandr-devel \

luajit-devel \

mbedtls \

swig \

python3-devel \

libxss-dev

Building and installing OBS:

If building with browser source:

wget https://cdn-fastly.obsproject.com/downloads/cef\_binary\_5060\_linux64.tar.bz2

tar -xjf ./cef\_binary\_5060\_linux64.tar.bz2

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DCMAKE\_INSTALL\_PREFIX=/usr -DENABLE\_BROWSER=ON -DCEF\_ROOT\_DIR="../../cef\_binary\_5060\_linux64" -DENABLE\_AJA=OFF ..

make -j4

sudo make install

If building without browser source:

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DLINUX\_PR=1 -DCMAKE\_INSTALL\_PREFIX=/usr -DBUILD\_BROWSER=OFF -DENABLE\_AJA=OFF ..

make -j4

sudo make install

# **Linux portable mode (all distros)**

Please note that you need to install the build dependencies for your distribution before following this steps. See above.

You can build in portable mode on Linux, which installs all the files to an isolated directory.

If building with browser source:

wget https://cdn-fastly.obsproject.com/downloads/cef\_binary\_5060\_linux64.tar.bz2

tar -xjf ./cef\_binary\_5060\_linux64.tar.bz2

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DLINUX\_PORTABLE=ON -DCMAKE\_INSTALL\_PREFIX="${HOME}/obs-studio-portable" -DENABLE\_BROWSER=ON -DCEF\_ROOT\_DIR="../../cef\_binary\_5060\_linux64" -DENABLE\_AJA=OFF ..

make -j4 && make install

If building without browser source:

git clone --recursive https://github.com/obsproject/obs-studio.git

cd obs-studio

mkdir build && cd build

cmake -DLINUX\_PORTABLE=ON -DCMAKE\_INSTALL\_PREFIX="${HOME}/obs-studio-portable" -DENABLE\_BROWSER=OFF -DENABLE\_AJA=OFF ..

make -j4 && make install

After that, you should have a portable install in ~/obs-studio-portable. Change to bin/64bit or bin/32bit and then simply run: ./obs