Obs-studio 安装

# **Install Instructions**

If you want to develop for OBS, please visit our [Discord](https://obsproject.com/discord) and get to know the devs or have questions answered!

Also, if there is something in this guide you want to change/improve on, it is recommended that you talk about it with the devs in Discord or IRC first.

Please note that any install directions/packages for Linux/FreeBSD distributions listed as *Unofficial* means that they are community provided, and any support for those packages should be directed at the appropriate distro/package maintainers.

### Table of Contents:

* [Windows](https://obsproject.com/wiki/Install-Instructions" \l "windows)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "windows-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "windows-build-directions)
* [macOS](https://obsproject.com/wiki/Install-Instructions" \l "macos)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "macos-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "macos-build-directions)
  + [Xcode Project](https://obsproject.com/wiki/Install-Instructions" \l "macos-xcode-project)
* [Linux](https://obsproject.com/wiki/Install-Instructions" \l "linux)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "linux-install-directions)
    1. [Ubuntu/Mint](https://obsproject.com/wiki/Install-Instructions" \l "ubuntumint-installation)
    2. [Arch Linux (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "arch-linux-installation-unofficial)
    3. [Manjaro (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "manjaro-installation-unofficial)
    4. [Fedora (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "fedora-installation-unofficial)
    5. [OpenMandriva Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "openmandriva-installation-unofficial)
    6. [Solus Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "solus-installation-unofficial)
    7. [openSUSE (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "opensuse-installation-unofficial)
    8. [Gentoo (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "gentoo-installation-unofficial)
    9. [NixOS (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "nixos-installation-unofficial)
    10. [UOS/Deepin (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "uosdeepin-installation-unofficial)
    11. [Debian/LMDE Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "debianlmde-installation-unofficial)
    12. [Void Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "void-installation-unofficial)
    13. [snappy (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "snappy-installation-unofficial)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "linux-build-directions)
    1. [Red Hat-Based](https://obsproject.com/wiki/Install-Instructions" \l "red-hat-based-build-directions)
    2. [Fedora](https://obsproject.com/wiki/Install-Instructions" \l "fedora-build-directions)
    3. [Debian-based](https://obsproject.com/wiki/Install-Instructions" \l "debian-based-build-directions)
    4. [openSUSE](https://obsproject.com/wiki/Install-Instructions" \l "opensuse-build-directions)
    5. [Linux portable mode (all distros)](https://obsproject.com/wiki/Install-Instructions" \l "linux-portable-mode-all-distros)
* [FreeBSD](https://obsproject.com/wiki/Install-Instructions" \l "freebsd)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "freebsd-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "freebsd-build-directions)

# **Windows**

### Windows Install Directions:

Pre-built Windows versions can be found here: <https://github.com/obsproject/obs-studio/releases/>

The full .exe installer and .zip contains OBS Studio 32bit, 64bit, Browser Source, and Intel® RealSense™ plugin. You will be prompted during install for the Browser Source and RealSense plugin to be installed if using the .exe installer, otherwise the components are included in the .zip.

The small .exe installer contains the base OBS Studio 32bit, 64bit, Intel® RealSense™ plugin, but does not contain the Browser Source plugin.

NOTE: If using the .zip method for either the full or small install and installing to a non-standard program location (i.e. outside Program Files), you will need to add the security group ALL APPLICATION PACKAGES to have full control over the main OBS Studio directory and sub-directories. Certain features may not function properly without these security rights (primarily, the ability to use game capture on UWP apps).

### Windows Build Directions:

****Requirements for building OBS on Windows****

* + Development packages of FFmpeg, x264, cURL, and mbedTLS.
    - Pre-built Windows dependencies for VS2019 can be found here:
      * VS2019: <https://obsproject.com/downloads/dependencies2019.zip>
  + [Qt5](http://www.qt.io/) (Grab the MSVC package for your version of Visual Studio)
    - We currently deploy with Qt 5.15.2
  + CEF Wrapper 3770 ([x64](https://cdn-fastly.obsproject.com/downloads/cef_binary_75.1.16+g16a67c4+chromium-75.0.3770.100_windows64_minimal.zip), [x86](https://cdn-fastly.obsproject.com/downloads/cef_binary_75.1.16+g16a67c4+chromium-75.0.3770.100_windows32_minimal.zip))
  + Windows version of [CMake](http://www.cmake.org/) (3.16 or higher, latest preferred)
  + Windows version of [Git](https://git-scm.com/download/win) (Git binaries must exist in path)
  + [Visual Studio 2019 (recommended)](https://visualstudio.microsoft.com/vs/)
    - Windows 10 SDK (minimum 10.0.19041.0). [Latest SDK](https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/)

****Installation Procedure****

Clone the repository and ****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****.

Create one or more of the following subdirectories within the cloned repository for building: release, debug, and build (suffixed with or without 32/64 to specify architecture). They are excluded from the repo in .gitignore for the sake of building, so they are safe to create an use within the repository base directory.

Run cmake-gui, and set the following fields:

* + - In "where is the source code", enter in the repo directory (example: D:/obs).
    - In "where to build the binaries", enter the repo directory path with the 'build' subdirectory (example: D:/obs/build).

Set the following variables. You can set them in cmake-gui, or you can set them as Windows Environment Variables to persist across configurations.

****Required****

DepsPath

DepsPath is the path to folder containing the dependencies, not including Qt. Set this to the win32 or win64 directory from the Pre-built dependencies package that you downloaded earlier.  
For example, if you extracted the dependencies .zip to C:\obs-deps, DepsPath should be one of these:

* + - * + C:\obs-deps\win32
        + C:\obs-deps\win64

If you want to specify both 32 and 64 bit dependencies to avoid changing the variable between builds, you can instead set DepsPath32 and DepsPath64 like so:

* + - * + DepsPath32: C:\obs-deps\win32
        + DepsPath64: C:\obs-deps\win64

QTDIR

QTDIR is the path to the Qt install directory. The OBS UI is built by default, which requires Qt. Set the CMake boolean variable DISABLE\_UI to TRUE if you don't want the GUI and this is no longer required. Can be optionally suffixed with 32 or 64 to specify target arch.

****NOTE****: Make sure to download Qt prebuilt components for your version of MSVC (32 or 64 bit).

Example Qt directories you would use if you installed Qt to D:\Qt would usually look something like this:

* + - * + QTDIR32=D:\Qt\5.15.2\msvc2019 (32-bit)
        + QTDIR64=D:\Qt\5.15.2\msvc2019\_64 (64-bit)

CEF\_ROOT\_DIR

CEF\_ROOT\_DIR is the path to an extracted CEF Wrapper. We provide a custom prebuilt wrapper to simplify the build process. This custom build includes access to hardware acceleration and additional codecs. ****This enables Browser Source and Custom Browser Docks.****

****Be sure to also enable the CMake flag**BUILD\_BROWSER**otherwise this will do nothing****

****Optional****

* + - * VIRTUALCAM\_GUID - Set to any random GUID value. This must be set to build the Virtual Camera features.

(If these components below share the same directory as DepsPath, they do not need to be individually specified.)

* + - * FFmpegPath - Path to just FFmpeg include directory.
      * x264Path - Path to just x264 include directory.
      * curlPath - Path to just cURL include directory.

****INFORMATIONAL NOTE****: Search paths and search order for base dependency library/binary files, relative to their include directories:

Library files

* + - * ../lib
      * ../lib32 (if 32bit)
      * ../lib64 (if 64bit)
      * ./lib
      * ./lib32 (if 32bit)
      * ./lib64 (if 64bit)

Binary files:

* + - * ../bin
      * ../bin32 (if 32bit)
      * ../bin64 (if 64bit)
      * ./bin
      * ./bin32 (if 32bit)
      * ./bin64 (if 64bit)

In cmake-gui, press 'Configure' and select the generator that fits to your installed VS Version:  
Visual Studio 16 2019, ****or their 64bit equivalents**** if you want to build the 64bit version of OBS

* + - NOTE: If you need to change your dependencies from a build already configured, you will need to uncheck COPIED\_DEPENDENCIES and run Configure again.

If you did not set up Environment Variables earlier you can now configure the DepsPath and if necessary the x264, FFmpeg and cUrl path in the cmake-gui.

In cmake-gui, press 'Generate' to generate Visual Studio project files in the 'build' subdirectory.

Open obs-studio.sln from the subdirectory you specified under "where to build the binaries" (e.g. D:/obs/build) in Visual Studio (or click the "Open Project" button from within cmake-gui).

The project should now be ready to build and run. All required dependencies should be copied on compile and it should be a fully functional build environment. The output is built in the 'rundir/[build type]' directory of your 'build' subdirectory.

If you want to use the Virtual Camera from this build, you will need to run its install script. If you already have a standard OBS Studio installation, you will need to uninstall its Virtual Camera first.

To uninstall an OBS Virtual Camera:

* + 1. Close any applications that were using the OBS Virtual Camera.
    2. In the obs-studio installation directory, run data\obs-plugins\win-dshow\virtualcam-uninstall.bat as administrator.

To install an OBS Virtual Camera:

* + 1. In the obs-studio installation directory (for Visual Studio builds, this is '[build dir]/rundir/[build type]'), run data\obs-plugins\win-dshow\virtualcam-install.bat as administrator.

Don't forget to uninstall your build's virtual camera before cleaning/deleting your build files.

****Integrating clang-format into Visual Studio****

* + clang-format is required for pull requests, and OBS uses a newer version than the one VS2019 ships with.
  + Download and install [LLVM 8.0.0](http://releases.llvm.org/)
  + Run VS, and go to Tools -> Options...
    - Text Editor -> C/C++ -> Code Style -> Formatting -> General
      * Enable "Use custom clang-format.exe" and enter the file name. For example:
        + C:\Program Files\LLVM\bin\clang-format.exe
  + The default command for formatting a document (Edit.FormatDocument) is Ctrl+K, Ctrl+D.

# **Install Instructions**

If you want to develop for OBS, please visit our [Discord](https://obsproject.com/discord) and get to know the devs or have questions answered!

Also, if there is something in this guide you want to change/improve on, it is recommended that you talk about it with the devs in Discord or IRC first.

Please note that any install directions/packages for Linux/FreeBSD distributions listed as *Unofficial* means that they are community provided, and any support for those packages should be directed at the appropriate distro/package maintainers.

### Table of Contents:

* [Windows](https://obsproject.com/wiki/Install-Instructions" \l "windows)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "windows-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "windows-build-directions)
* [macOS](https://obsproject.com/wiki/Install-Instructions" \l "macos)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "macos-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "macos-build-directions)
  + [Xcode Project](https://obsproject.com/wiki/Install-Instructions" \l "macos-xcode-project)
* [Linux](https://obsproject.com/wiki/Install-Instructions" \l "linux)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "linux-install-directions)
    1. [Ubuntu/Mint](https://obsproject.com/wiki/Install-Instructions" \l "ubuntumint-installation)
    2. [Arch Linux (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "arch-linux-installation-unofficial)
    3. [Manjaro (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "manjaro-installation-unofficial)
    4. [Fedora (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "fedora-installation-unofficial)
    5. [OpenMandriva Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "openmandriva-installation-unofficial)
    6. [Solus Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "solus-installation-unofficial)
    7. [openSUSE (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "opensuse-installation-unofficial)
    8. [Gentoo (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "gentoo-installation-unofficial)
    9. [NixOS (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "nixos-installation-unofficial)
    10. [UOS/Deepin (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "uosdeepin-installation-unofficial)
    11. [Debian/LMDE Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "debianlmde-installation-unofficial)
    12. [Void Installation (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "void-installation-unofficial)
    13. [snappy (Unofficial)](https://obsproject.com/wiki/Install-Instructions" \l "snappy-installation-unofficial)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "linux-build-directions)
    1. [Red Hat-Based](https://obsproject.com/wiki/Install-Instructions" \l "red-hat-based-build-directions)
    2. [Fedora](https://obsproject.com/wiki/Install-Instructions" \l "fedora-build-directions)
    3. [Debian-based](https://obsproject.com/wiki/Install-Instructions" \l "debian-based-build-directions)
    4. [openSUSE](https://obsproject.com/wiki/Install-Instructions" \l "opensuse-build-directions)
    5. [Linux portable mode (all distros)](https://obsproject.com/wiki/Install-Instructions" \l "linux-portable-mode-all-distros)
* [FreeBSD](https://obsproject.com/wiki/Install-Instructions" \l "freebsd)
  + [Install](https://obsproject.com/wiki/Install-Instructions" \l "freebsd-install-directions)
  + [Build from source](https://obsproject.com/wiki/Install-Instructions" \l "freebsd-build-directions)

# **Windows**

### Windows Install Directions:

Pre-built Windows versions can be found here: <https://github.com/obsproject/obs-studio/releases/>

The full .exe installer and .zip contains OBS Studio 32bit, 64bit, Browser Source, and Intel® RealSense™ plugin. You will be prompted during install for the Browser Source and RealSense plugin to be installed if using the .exe installer, otherwise the components are included in the .zip.

The small .exe installer contains the base OBS Studio 32bit, 64bit, Intel® RealSense™ plugin, but does not contain the Browser Source plugin.

NOTE: If using the .zip method for either the full or small install and installing to a non-standard program location (i.e. outside Program Files), you will need to add the security group ALL APPLICATION PACKAGES to have full control over the main OBS Studio directory and sub-directories. Certain features may not function properly without these security rights (primarily, the ability to use game capture on UWP apps).

### Windows Build Directions:

****Requirements for building OBS on Windows****

* + Development packages of FFmpeg, x264, cURL, and mbedTLS.
    - Pre-built Windows dependencies for VS2019 can be found here:
      * VS2019: <https://obsproject.com/downloads/dependencies2019.zip>
  + [Qt5](http://www.qt.io/) (Grab the MSVC package for your version of Visual Studio)
    - We currently deploy with Qt 5.15.2
  + CEF Wrapper 3770 ([x64](https://cdn-fastly.obsproject.com/downloads/cef_binary_75.1.16+g16a67c4+chromium-75.0.3770.100_windows64_minimal.zip), [x86](https://cdn-fastly.obsproject.com/downloads/cef_binary_75.1.16+g16a67c4+chromium-75.0.3770.100_windows32_minimal.zip))
  + Windows version of [CMake](http://www.cmake.org/) (3.16 or higher, latest preferred)
  + Windows version of [Git](https://git-scm.com/download/win) (Git binaries must exist in path)
  + [Visual Studio 2019 (recommended)](https://visualstudio.microsoft.com/vs/)
    - Windows 10 SDK (minimum 10.0.19041.0). [Latest SDK](https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/)

****Installation Procedure****

Clone the repository and ****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****.

Create one or more of the following subdirectories within the cloned repository for building: release, debug, and build (suffixed with or without 32/64 to specify architecture). They are excluded from the repo in .gitignore for the sake of building, so they are safe to create an use within the repository base directory.

Run cmake-gui, and set the following fields:

* + - In "where is the source code", enter in the repo directory (example: D:/obs).
    - In "where to build the binaries", enter the repo directory path with the 'build' subdirectory (example: D:/obs/build).

Set the following variables. You can set them in cmake-gui, or you can set them as Windows Environment Variables to persist across configurations.

****Required****

DepsPath

DepsPath is the path to folder containing the dependencies, not including Qt. Set this to the win32 or win64 directory from the Pre-built dependencies package that you downloaded earlier.  
For example, if you extracted the dependencies .zip to C:\obs-deps, DepsPath should be one of these:

* + - * + C:\obs-deps\win32
        + C:\obs-deps\win64

If you want to specify both 32 and 64 bit dependencies to avoid changing the variable between builds, you can instead set DepsPath32 and DepsPath64 like so:

* + - * + DepsPath32: C:\obs-deps\win32
        + DepsPath64: C:\obs-deps\win64

QTDIR

QTDIR is the path to the Qt install directory. The OBS UI is built by default, which requires Qt. Set the CMake boolean variable DISABLE\_UI to TRUE if you don't want the GUI and this is no longer required. Can be optionally suffixed with 32 or 64 to specify target arch.

****NOTE****: Make sure to download Qt prebuilt components for your version of MSVC (32 or 64 bit).

Example Qt directories you would use if you installed Qt to D:\Qt would usually look something like this:

* + - * + QTDIR32=D:\Qt\5.15.2\msvc2019 (32-bit)
        + QTDIR64=D:\Qt\5.15.2\msvc2019\_64 (64-bit)

CEF\_ROOT\_DIR

CEF\_ROOT\_DIR is the path to an extracted CEF Wrapper. We provide a custom prebuilt wrapper to simplify the build process. This custom build includes access to hardware acceleration and additional codecs. ****This enables Browser Source and Custom Browser Docks.****

****Be sure to also enable the CMake flag**BUILD\_BROWSER**otherwise this will do nothing****

****Optional****

* + - * VIRTUALCAM\_GUID - Set to any random GUID value. This must be set to build the Virtual Camera features.

(If these components below share the same directory as DepsPath, they do not need to be individually specified.)

* + - * FFmpegPath - Path to just FFmpeg include directory.
      * x264Path - Path to just x264 include directory.
      * curlPath - Path to just cURL include directory.

****INFORMATIONAL NOTE****: Search paths and search order for base dependency library/binary files, relative to their include directories:

Library files

* + - * ../lib
      * ../lib32 (if 32bit)
      * ../lib64 (if 64bit)
      * ./lib
      * ./lib32 (if 32bit)
      * ./lib64 (if 64bit)

Binary files:

* + - * ../bin
      * ../bin32 (if 32bit)
      * ../bin64 (if 64bit)
      * ./bin
      * ./bin32 (if 32bit)
      * ./bin64 (if 64bit)

In cmake-gui, press 'Configure' and select the generator that fits to your installed VS Version:  
Visual Studio 16 2019, ****or their 64bit equivalents**** if you want to build the 64bit version of OBS

* + - NOTE: If you need to change your dependencies from a build already configured, you will need to uncheck COPIED\_DEPENDENCIES and run Configure again.

If you did not set up Environment Variables earlier you can now configure the DepsPath and if necessary the x264, FFmpeg and cUrl path in the cmake-gui.

In cmake-gui, press 'Generate' to generate Visual Studio project files in the 'build' subdirectory.

Open obs-studio.sln from the subdirectory you specified under "where to build the binaries" (e.g. D:/obs/build) in Visual Studio (or click the "Open Project" button from within cmake-gui).

The project should now be ready to build and run. All required dependencies should be copied on compile and it should be a fully functional build environment. The output is built in the 'rundir/[build type]' directory of your 'build' subdirectory.

If you want to use the Virtual Camera from this build, you will need to run its install script. If you already have a standard OBS Studio installation, you will need to uninstall its Virtual Camera first.

To uninstall an OBS Virtual Camera:

* + 1. Close any applications that were using the OBS Virtual Camera.
    2. In the obs-studio installation directory, run data\obs-plugins\win-dshow\virtualcam-uninstall.bat as administrator.

To install an OBS Virtual Camera:

* + 1. In the obs-studio installation directory (for Visual Studio builds, this is '[build dir]/rundir/[build type]'), run data\obs-plugins\win-dshow\virtualcam-install.bat as administrator.

Don't forget to uninstall your build's virtual camera before cleaning/deleting your build files.

****Integrating clang-format into Visual Studio****

* + clang-format is required for pull requests, and OBS uses a newer version than the one VS2019 ships with.
  + Download and install [LLVM 8.0.0](http://releases.llvm.org/)
  + Run VS, and go to Tools -> Options...
    - Text Editor -> C/C++ -> Code Style -> Formatting -> General
      * Enable "Use custom clang-format.exe" and enter the file name. For example:
        + C:\Program Files\LLVM\bin\clang-format.exe
  + The default command for formatting a document (Edit.FormatDocument) is Ctrl+K, Ctrl+D.