github.com /AppImage/AppImageKit/wiki/FUSE

FUSE

4-5 minutes

AppImages require FUSE version 2 to run. Filesystem in Userspace (FUSE) is a system that lets non-root users mount filesystems.

Install FUSE

Many distributions have a working FUSE setup out-of-the-box. However if it is not working for you, you may need to install and configure FUSE manually.

For example, on **Ubuntu (>= 24.04)**:

```
sudo add-apt-repository universe
sudo apt install libfuse2t64
```

Note: In Ubuntu 24.04, the `libfuse2` package was renamed to `libfuse2t64`.

For example, on **Ubuntu (>= 22.04)**:

```
sudo add-apt-repository universe
sudo apt install libfuse2
```

Warning: While `libfuse2` is OK, do not install the `fuse` package as of 22.04 or you may break your system. If the fuse package did break your system, you can recover as described here.

For example, on **Ubuntu (<= 21.10)**:

```
sudo apt install fuse libfuse2
sudo modprobe fuse
sudo groupadd fuse

user="$(whoami)"
sudo usermod -a -G fuse $user
```

For example, on **openSUSE**:

```
sudo zypper install fuse libfuse2
```

In order to use `fusermount` on OpenSUSE with the default (?) "secure" file permission settings (see `/etc/permissions.secure`), your login needs to be part of the `trusted` group. To add yourself, run

```
sudo usermod -a -G trusted `whoami`
```

Then log out and log in for the change to take effect.

For example, on **Fedora**:

```
dnf install fuse fuse-libs
```

For example, on **CentOS/RHEL**:

```
yum --enablerepo=epel -y install fuse-sshfs # install from EPEL
user="$(whoami)"
usermod -a -G fuse "$user"
```

For example, on Armbian 64-bit systems (e.g., for the Pine64) you need to install 32-bit libfuse2 in order to run 32-bit AppImages such as the MuseScore one:

```
sudo apt install libfuse2:armhf
```

For example, on **Clear Linux OS**:

This may be a bug, please see https://github.com/clearlinux/distribution/issues/273

```
sudo su
mkdir -p /etc/modules-load.d/
echo "fuse" > /etc/modules-load.d/fuse.conf
reboot
```

For example, on **Arch Linux**:

If you are seeing "fusermount: mount failed: Operation not permitted"

```
sudo chmod u+s `which fusermount`
```

Chrome OS, Chromium OS, Crostini:

FUSE is not operational out of the box, but installation is simple after version 73:

check https://bugs.chromium.org/p/chromium/issues/detail?id=841787 for details

Fallback

If you don't want to install FUSE, you can either mount or extract the AppImage.

type-2 AppImage

To extract the contents of the AppImage, simply run the AppImage with `--appimage-extract`.

type-1 AppImage

If the above does not work, you may still have an older type-1 AppImage. To mount the AppImage and run the application, simply run

```
sudo mount -o loop Some.AppImage /mnt
/mnt/AppRun
```

A type-1 AppImage is an ISO, so

```
sudo apt install libarchive-tools # Or any other method to get `bsdtar`
mkdir AppDir
cd AppDir
bsdtar xfp /home/me/Downloads/Some.AppImage
./AppRun
```

also works.

Docker

When running an AppImage from a Docker container you will get the following error:

```
fuse: failed to open /dev/fuse: Operation not permitted
Could not mount AppImage
Please see https://github.com/probonopd/AppImageKit/wiki/FUSE
```

You'll often hear "oh, just add these arguments to `docker run --cap-add SYS_ADMIN --cap-add MKNOD --device /dev/fuse:mrw` and it'll work", but that is considered to be insecure.

Instead, just use the `--appimage-extract-and-run` parameter to the AppImage in your build script:

```
[...]
./appimagetool-*.AppImage --appimage-extract-and-run ...
[...]
```

You can also make `appimagetool` do this using `export APPIMAGE_EXTRACT_AND_RUN=1`

Note: `appimagetool-*.AppImage` can be extracted starting with release version 9

If you want to decide whether to use the AppImage directly or extracted depending on whether you're in a container or not, for example in a build script, you can combine this with some detection code.

More troubleshooting information

https://docs.appimage.org/user-guide/troubleshooting/fuse.html

On https://copy.sh/v86/?profile=archlinux , you have to get the .so files from .deb files.